IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 3, 2019, accepted October 20, 2019, date of publication November 4, 2019, date of current version November 21, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2950955

"Digital Twins" for Highly Customized Electronic Devices – Case Study on a Rework Operation

RAFAŁ CUPEK[1], MAREK DREWNIAK[2], ADAM ZIEBINSKI[1], AND MARCIN FOJCIK[3]
[1]Institute of Informatics, Silesian University of Technology, 44-100 Gliwice, Poland
[2]Automation Department, AIUT Sp. Z o.o., 44-109 Gliwice, Poland
[3]Department of Electrical Engineering, Western Norway University of Applied Sciences, 6812 Førde, Norway

Corresponding author: Rafał Cupek (rcupek@polsl.pl)

This work was supported in part by the Polish National Centre of Research and Development from the Project (Knowledge integrating shop floor management system supporting preventive and predictive maintenance services for automotive polymorphic production framework) under Agreement POIR.01.02.00-00-0307/16-00, and in part by the B+R sector programmes of Intelligent Development operational programme from 2014–2020 through the European Regional Development Fund.

ABSTRACT The ongoing changes in manufacturing require that new information models for industrial computer systems be developed and applied. This paper describes a concept for the material model as a "digital twin" for producing highly customised, smart electronic devices. The scope of the research is the transformation of the models that are typical for the currently used automation pyramid approach to Reference Architecture Models for Industry 4.0 (RAMI4.0). The ISA95 standard is used as the modelling tool and Open Production Connectivity Unified Architecture (OPC UA) as the communication middleware. The presented use case focuses on a rework operation that is performed during the short series production of highly customised electronic devices that are produced by the Aiut company. The paper focuses on the transformation from the static architecture of Manufacturing Execution Systems to flexible and dynamic information models.

INDEX TERMS Common information model (computing), computer integrated manufacturing, flexible manufacturing systems, intelligent manufacturing systems.

I. INTRODUCTION

The fourth industrial revolution is associated with agile and collective manufacturing [1], whose enabling technologies are the digitalisation and common use of individualised ICT (Information and Communication Technology) services that can easily and automatically be adjusted to fit a specific product and production needs [2]. The next generation of smart manufacturing systems will benefit from the transparent information models and universal communication middleware that are enabling factors for flexible manufacturing [3].

By increasing the amount of automation at multiple levels within a factory and across the enterprise, the concept of cyber-physical systems, in which computation and networking technologies interact with physical systems, have made strides into manufacturing systems. The cyber-physical manufacturing systems enable higher productivity and higher quality as well as lower costs [4].

The associate editor coordinating the review of this manuscript and approving it for publication was Okyay Kaynak.

The fourth industrial revolution can be symbolised by a machine that is connected to the market that is able to perform one-piece flow production that fits into the individual requirements connected with a given production step. This approach includes the idea of a "digital twin" [5] – a digital representation of the physical items that are necessary for the new generation of manufacturing in which physical production systems, virtual factories, ICT services and digital data are merged.

Unfortunately, the vast majority of contemporary IT solutions that are used in manufacturing are based on a centralised architecture [6], which forces the information models for individual applications to meet the paradigm of vertical communication. The data exchange services, communication interfaces and information formats are rigidly determined when the system is created. Moreover, information models and communication interfaces do not fit together at different levels of the manufacturing system. Such an architectural model is characterised by a lack of flexibility. Each change in the scope of the business model or the production technology forces subsequent changes in the system's architecture.

On the one hand, the cost of making these changes is high, while on the other hand, there is a significant risk of a loss of information coherence as well as the risk of functional errors. Moreover, the dependencies between the data and the services that are implemented by the system are sewn into the application software.

Resource virtualisation or the creation of digital twins is a key enabling technology in the generic smart factory architecture. In [7], a test-driven resource virtualisation process was proposed as a recommendation for industry to adopt in order to create digital twins for their smart factory solutions, especially with the assistance of a developed web interface, thereby making the entire process a virtualisation. Cyber-Physical Systems and Digital Twin technologies can be used to build the interconnection and interoperability of a physical shop floor and the corresponding cybershop floor [8] and will form the basis for shop floors to march towards smart manufacturing. Therefore, contemporary manufacturing systems are based on engineering knowledge that is hidden and is unavailable to other production systems.

In order to make information more transparent and to reduce the complexity during the engineering process an ontology-based approach can be used [9], which allows process engineers to deal with fewer technical details so that they can predominantly focus on the design of the actual product that is to be produced. Another approach can be based on knowledge discovery methods such as the application of a training approach that is based on a hidden Markov model that is based on historical data in order to allocate them autonomously for manufacturing tasks [10].

This research is focused on a transparent material model that can support the interoperable ICT services that are used in the new generation of discrete manufacturing systems. One of the most important ICT operations in manufacturing is material tracking, which should be performed from the beginning of a material's life cycle (as raw materials) through the semi-products to the end products that will be used by customers. The authors focus on the production of electronic devices, particularly on the components of smart gas meters. A material model should be flexible enough to be adjusted easily at every step of production as well as later during product maintenance or recycling. The analysis starts from a very generic material model that was defined by RAMI4.0 and ISA95 standards and ends with an architectural design dedicated for the considered use case – the production of smart gas meters that is performed by Aiut Ltd. The model forms the backbone for the services that are required in the shop-floor management system.

Despite advanced production control methods, errors in the short series production of electronic devices can be quite high. According to Aiut's quality management department, the error rate can vary from a few to several percent of production. They can be caused by the incorrect quality of the input materials, the wrong choice of materials, the incorrect technology being used and finally production staff errors. Such problems are typical in the case of the short series production of electronic devices [11]. In many kinds of discrete manufacturing, a rework operation that is dedicated for fixing product errors is still an important part of the production process. The presented use case focuses on a rework operation that utilises information about the errors that were detected during production. In the case of the many different devices and different device versions that are processed by a rework station, only clear and unambiguous information can be the basis for efficient rework services. On the other hand, transparent data models are necessary for product improvement and/or production optimisation.

The main contribution of this paper is the case-based presentation of a new approach for information modelling that is dedicated for the new generation of manufacturing systems. Since the new models that are presented were applied in an existing manufacturing system of smart electronic devices production, its implementation will focus on a number of constraints that are typical in the case of the previous generation of industrial computer systems. The presented solution fully supports the new flexible materials models, which can be used in the case of the shorts series production of advanced electronic devices, but is also proposed as a bridge between classical Manufacturing Execution Systems and the classical Enterprise Resource Management approach and the new modelling approach that was developed under RAMI4.0 models and two parts that follow the OPC UA and ISA95 standards.

The rest of this paper is organised as follows: the second section presents main features of Reference Architecture for Industry4.0 (RAMI4.0) and two of its components: the ISA95 (IEC 62264) standard that defines the ontology for MES (Manufacturing Execution Systems) and the object-oriented communication middleware OPC UA (IEC 62541). Section three presents the use case – the production of smart gas meters with a focus on the information that is used during the rework operation. The practical design for the flexible material model is presented in the fourth section and the conclusions are presented in section five. The authors believe that the most important contribution of the paper is (i) a use case-based analysis to create a material model that is compatible with RAMI4.0, ISA95 and OPC UA. Since ISA 95 is defined as the general model described in UML, it does not give any indication of how to implement it in an actual system. The authors use ISA95 in accordance with the RAMI4.0 and OPC UA; (ii) an ontology-based approach for communication middleware that is dedicated for material information modelling. OPC UA permits advanced data models that can be applied in industry to be created, but it does not define any ready-to-use patterns. The authors show how to create an ontology-based material information model that is based on OPC UA and fits both the generic standard and a given application field; (iii) an information model that takes into account the M2M (Machine-to-Machine) communication and communication with a human operator. The rework process requires individual actions that are to be taken that are tailored to the specificity of the production error.

II. MATERIAL MODELLING IN SMART MANUFACTURING

A. RAMI4.0 4.0 THE SUCCESOR OF THE AUTOMATION PYRAMID

A Reference Architecture Model for Industry 4.0 (RAMI4.0) is a high-level schema that focuses on integrating manufacturing services. It systematises the architecture and interaction between components and enables digitalisation in the new generation of manufacturing systems [12]. Nowadays, RAMI4.0 replaces the previous model, which was called an automation pyramid. The automation pyramid was created in the late 1990s and is still commonly used to describe industrial computer systems.

1) THE AUTOMATION PYRAMID

An automation pyramid is composed of five layers [13] that are responsible for processing, converting and exchanging information (Fig. 1). From bottom to top: (0) Sensors and actuators; (1) Control systems; (2) Visualisation and supervisory control; (3) Manufacturing Execution Systems and (4) Management support systems. The automation pyramid enables the description of an industrial computer system to be simplified. The complexity of a system that is composed of multiple different kinds of sensors and actuators, PLCs (Programmable Logic Controllers), industrial computers, complex DCS (Distributed Control Systems), SCADA (Supervisory Control and Data Acquisition) systems and industrial databases that are called Historians is structured by this model and the technical solutions that are used for many systems in industry follow it.

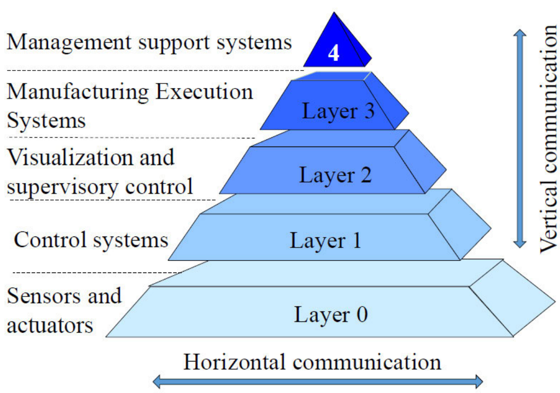

FIGURE 1. Model of an automation pyramid.

In the automation pyramid, MES (Manufacturing Execution Systems) and ERP (Enterprise Resource Planning) are located in the upper layers of the model. This reflects the static hierarchy of manufacturing services, which reduces flexibility and increases the cost of adapting the system to new conditions. This problem has been found by a number of advisory groups such as the German Platform Industrie-4.0, the NIST Institute (US National Institute of Standards and Technology) or the SMLC (Smart Manufacturing Leadership Coalition), which operates in South Korea [14].

2) RAMI4.0

RAMI4.0 is one of the new generations of high-level manufacturing models that was developed by Industrie-4.0 and that focuses on flexible production systems that support agile manufacturing. It structures the existing standards, finds any missing links between the standards and indicates the areas that require standardisation.

The generic 3-dimensional representation of RAMI4.0 is shown in Fig. 2. The vertical axis of RAMI4.0 models the hierarchy of services instead of the hierarchy of a system's components. Therefore, it can be used for modelling a system on different complexity levels. Specific ICT services can be presented as complete functional components and also as a part of a whole using a holonic approach. In such a representation, a service chain can be flexibly and dynamically created using an agent-based approach [15].

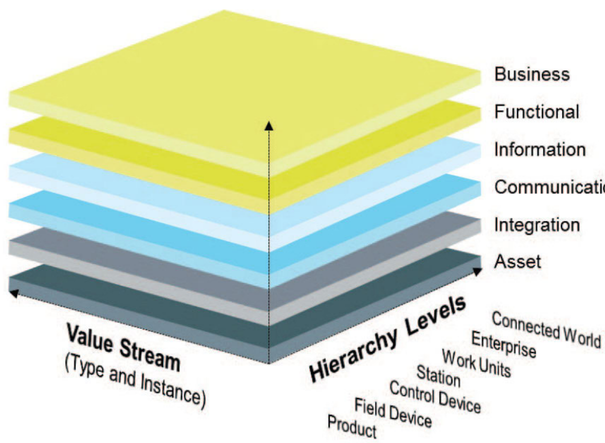

FIGURE 2. Reference architecture model for industry 4.0 (RAMI 4.0) [13].

3) THE VERTICAL AXIS OF RAMI4.0

The vertical axis of RAMI4.0 focuses on different aspects of the services by dividing them into six layers, which from the bottom to the top are:

The *Asset layer* represents reality, i.e. the asset that actually exists in the physical world (the physical materials of the considered case). It is this reality that has to be virtually represented in the layers above. This layer also covers other assets including the production staff, technology archives, documents, instructions, relevant software etc.;

The *Integration layer* represents the transition from the physical world to the information world. It describes the assets that are required to implement a function that represents or interacts with a resource. This layer is where the physical properties of the materials are measured, converted and stored. It provides a representation of the actual assets through information about the assets. It describes the physical parameters using the information from sensors, RFID readers, signal converters and the information that is provided by humans. It is also the place for the computer-aided control of the material properties and the source for generating the events and alarms that are connected with the assets;

The *Communication layer* describes the Industry 4.0-compliant access to the information and functions of a connected asset by other assets. In other words, it describes which data is used, where it is used and when it is distributed; standardised Industry4.0 communication using a uniform data format provides services that are based on a service-oriented architecture (SOA);

The *Information layer* describes the data that is used, generated or modified by the technical functionality of the asset. It is used by the runtime environment for (pre)processing the event, executing the rules, providing a formal description of the models and rules, the persisting data that is represented by the models. It ensures the integrity of the data and the consistent integration of different data, acquiring new, higher quality data (data, information, knowledge). It also provides structured data via the service interfaces, receiving events and transforming them into a suitable form for the data that is available to the functional layer and pre-processing the context;

The *Functional layer* describes the (logical) functions of the asset (technical functionality) with regard to its role in the Industry 4.0 system by providing a formal, digital description of the functions, being a platform for the horizontal integration of the runtime of different functions, and modelling the environment for the services and business processes *via* a runtime environment for the applications and technical functionality;

The *Business layer* describes the commercial view including the general organisational boundary conditions (such as order commissioning, general ordering conditions or regulatory provisions), the monetary conditions (price, availability of resources, discounts etc.) and the integrity of the functions in the value-added chain. In the case of a material model, this level is typically responsible for preparing the BOM (Bill of Materials) for input production and collecting the production records for the production output.

4) HORIZONTAL AXES OF RAMI4.0

The *left-hand horizontal axis* of RAMI 4.0 represents the Life Cycle and Value Stream and focuses on developing the product and the relevant production processes. It can serve as the cornerstone for product and process development because it enables management to be changed throughout all of the product and production system life cycle [16]. In the case of materials, it should reflect the continuous development of the materials that are used and produced. In the case of an object-oriented approach, the subsequent product versions should be expressed by a hierarchy of types. The inheritance mechanism for small product changes should lead to a direct representation of a parent-child relationship, while major changes should refer to the more basic types in the product hierarchy. *The left-hand horizontal axis* does not focus on specific physical components and their features (that are covered by the vertical axis) but reflects the development of the product and production process.

The *right-hand horizontal axis* represents the functional Hierarchy Levels and indicates the functionalities and responsibilities within factories/plants. It describes the functional classification of a production entity according to the various circumstances in Industry 4.0.

Although RAMI4.0 is a holistic model that covers the vast majority of manufacturing aspects, we have to remember that in the vast majority of industries, RAMI4.0 has to be applied in the existing production infrastructure and be adjusted to the technology of production that is currently being used. For this reason, new designs cannot be applied from scratch, but existing production systems have to evaluate converting to RAMI4.0. This evaluation should take into account the existing assets, the technologies being used, the management routines and operational procedures.

B. ISA95 MODELS FOR MANUFACTURING EXECUTION SYSTEMS

1) CIM

The concept of Computer Integrated Manufacturing (CIM) has its roots in the second half of the 1970s. The CIM model that was proposed by Harrington [17] was designed to combine separate IT systems such as systems that support the design and optimisation of process automation, systems that support the planning and control of production, systems for the preparation and optimisation of production processes, logistics systems, sales support systems and others. The CIM was to integrate information processing in the scale of the entire company as well as to permit information sharing between the systems that were being used to manage an enterprise and computer control [18].

This task required the preparation of a configurable, distributed IT platform that would not only allow many different IT systems to be connected but also enable the conversion of the information models that were defined and processed in separate domains [19]. The implementation of such an environment was initially limited by the availability of information technology and only further development in the field of ICT especially in the area of object-oriented communication middleware and service-based information models mean that the CIM idea is nowadays closer to being implemented than at any time.

2) MES

One of the key components of CIM are MES. The term Manufacturing Execution Systems (MES), which was proposed by Michael McClellan, means an on-line, integrated computer system that is used for the operational management of production [20]. MES are an interface that connect industrial control systems including supervisory control systems with the systems that support business management and business intelligence. The word execution refers to all of the activities that are related to the company that is performing the production operations. Nowadays, MES seems to be one of the cornerstones for the fourth industrial revolution. MES ensure the cooperation between the complex activities that combine various types of tools and methods and the resulting synergy. The information flow between the components of the system creates the added value that is associated with the optimisation of production [21].

MES have to cooperate with many different IT systems both inside and outside of a company, and therefore, they must be open and flexible [22]. The features that distinguish MES from other industrial IT systems are: (i) the real-time continuous communication with the production level, (ii) the online conversion of information models, (iii) the transformation of data that is presented in accordance with the business model into a form that is easily interpreted and executed by the industrial control systems and (iv) the conversion and processing of the data that is acquired from the control systems into a form that can be used by the other systems of the enterprise.

MES provide tools to adapt the various information models that are used in industry and provide services to transform raw production data into information that is relevant to operational business management. MES support operational management while management systems focus on the business operations. The decisions that are made at the level of management systems must be translated into the production activities that are aimed at implementing the business tasks of the enterprise. In this context, MES are extension of the management systems whose task is to separate and supervise the implementation of production tasks as well as to collect information on how the production tasks were performed.

During the time of mass production, MES were created as fixed sets of interfaces and services that reflected a given production model and strict manufacturing rules. In the time of agile manufacturing, MES have to follow different data models and offer services that can be easily adapted to frequent changes in low-volume production; otherwise MES will impede manufacturing rather than support it. In terms of the functional and architectural classes of information systems that are used by industry, MES combine industrial control systems and management systems. These two classes were created to perform different tasks, and therefore, they differ in terms of the ICT architecture that is used for their implementation. For discrete manufacturing, the control systems are mainly based on PLC (Programable Logic Controllers) architecture, while ERP are based on high performance computers and the batch processing principle.

3) ISA95

ISA95 (IEC 62264) is a commonly accepted industrial standard that defines the data flow and services for MES and can be used in all branches of manufacturing. ISA95 is maintained and developed by the MESA (Manufacturing Enterprise Solutions Association) and is based on the PURDUE model [23]. The core ideas of ISA95 originated in the 1990s when ISA (The Instrumentation, Systems and Automation Society) decided to develop a standard for integrating enterprise and control systems in order to reduce the risk, costs and errors connected with implementing the interfaces between such systems [24]. Nowadays, ISA95 is considered to be one of the keys enabling technologies for implementing manufacturing systems that are compatible with RAMI4.0 [16].

The activities described by ISA95 cover all four of the main parts that are required to perform manufacturing: production operations management, quality operations management, maintenance operations management and inventory operations management. The generic model for the exchange of MES services and information between them is presented in Fig. 3. The services are classified according to their adherence to the business, control and MES domains. The internal area of the model (within the dashed line) refers to the services that are performed by the control systems, the external area refers to services that are performed in business domain, the services situated on the border line refer to the MES activities.

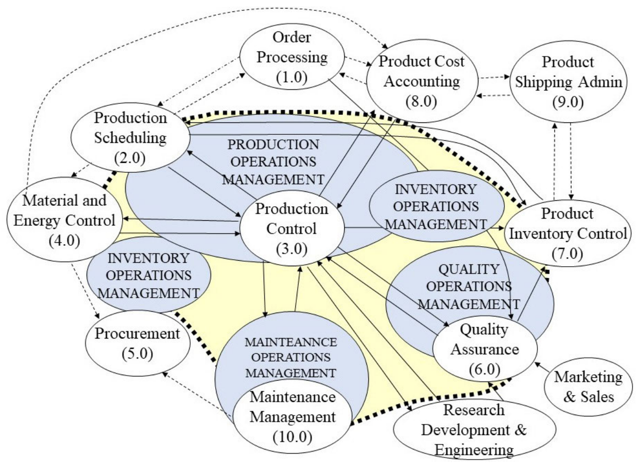

FIGURE 3. The Purdue manufacturing model that was adopted as the ISA95 standard [23].

The subsequent numbers indicate the ten main service groups and the arrows show information flow between them. Production starts from the acquisition of an order from a customer (step 1.0), after which a detailed production schedule is created by the Production Scheduling services (step 2.0), which are then performed by the Production Control (step 3.0). The Quality Assurance services (step 6.0) are performed during the production. In this work, the authors focus on a material model and therefore the Material Control (step 4.0), Production Control (step 3.0) and Product Inventory Control (step 7.0) are particularly important. Because the analysed rework operation utilises information about production errors, it exchanges information (bidirectionally) with Quality Assurance (6.0) services.

4) MATERIAL MODEL DEFINED BY ISA95

A rework station is a specific production stand that uses information about the material production process and especially the production errors that were detected. Only unambiguous information can ensure that the effective corrective actions are undertaken. On the other hand, a rework provides very valuable information about the technological problems, which may help to improve production technology. necessary information is organised under the MES services, which are classified according to the third part of the ISA95 [26] standard. The main application groups for the information services that use the material model are:

The ISA95 Material model is composed of the four mutually interconnected classes that are shown in Fig. 4. The Material Class and Material Definition are used to define the key features that describe the materials while the Material Lot and Material Sublot create a "digital twin" of the actual materials and bind them with the relevant model that is described by the Material Definition. The Material Definition can be grouped under many Material Class definitions. It reflects a situation in which a given material may have different purposes.

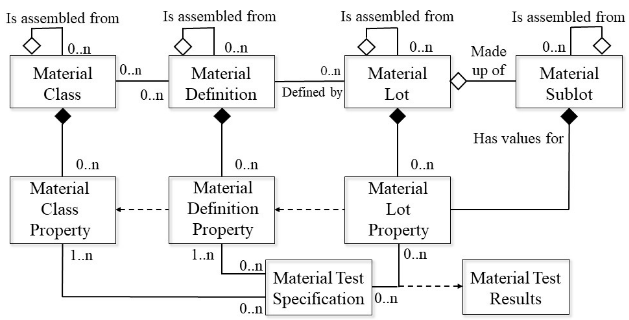

FIGURE 4. Material model defined by ISA95 [32].

The properties are used to indicate the physical features that have to be considered for a given material and its context of use. Each property is uniquely identified by its ID; it may have one or more descriptions and one or more values that are used to define a feature. Each material class may consist of many features that describe it, each of which can adopt the type and range of values that are defined by the Material Class and Material Definition. The model is recursive – the properties can be described by nested properties and the material class can be defined as a composition of the other material classes.

Such a representation allows each material to be described virtually using its classification, definition, collective and detailed description. The same schema is used for both the consumed and produced materials. Each Material Lot must be classified under one, and only one, Material Definition, which makes the description pattern unambiguous. The model also specifies how the material properties are tested and assessed. The assessment is performed according to the Material Test Specification procedure. A very fine description can be expressed by the Material Sublot, which can be created to model a smaller batch of materials or a single item. The description that is based on the Material Sublot can also be nested.

5) B2MML

The material model together with the equipment, personnel and asset models are the basic elements for the more complex definitions that are specified by ISA95. These basic components form the production and process segments, that are used for production scheduling, production tracking or performance analysis. Nowadays, ISA95 is well supported by ERP systems (for example SAP). The models that are defined under the ISA95 schemes that are given in the second part of the standard can be directly implemented for data exchange between MES and ERP systems using the B2MML (Business to Manufacturing Modeling Language) [25], which is an implementation tool for ISA95. The definitions given by ISA95 are represented by the relevant .xsd schemas that are used to create and verify the semantic correctness of a data representation. An example of a B2MML description for a Material Definition Property Type is shown in Fig. 5.

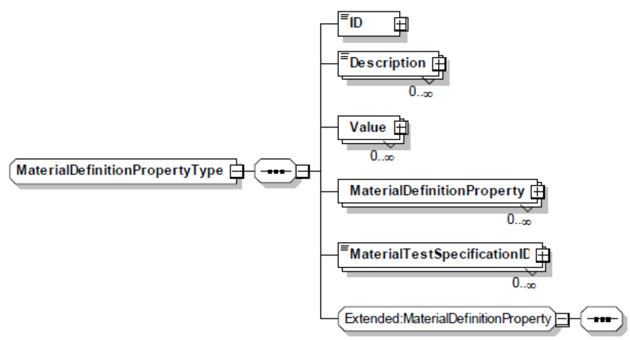

FIGURE 5. The B2MML format for a material definition property type [25].

The main advantage of an.xml-based representation is that the batch data exchange can be performed easily. Usually, the B2MML files that are sent by the ERP contain descriptions of all of the materials that are used for production as BOMs (Bills of Materials). The feedback information describes the properties of the batch of products that are described by the B2MML as the materials that were produced. B2MML-based communication has two main disadvantages: (i) access to the individual materials or their properties requires a search of the entire content of the B2MML file and (ii) the description of the material properties is a snapshot at the moment that it was created, but it does not allow for a presentation of how each property was changed during the production process.

Compared to the standard object-oriented design methods, ISA95 does not permit complex hierarchies that are based on an inheritance mechanism to be created, and, as a result, the models can hardly evolve. Moreover, B2MML can only support one point of view, which means that context-based descriptions are unavailable. In order to express the different aspects of a material, the models have to be complex or multiple material models have to be used. This leads to model redundancy and possible ambiguity between model parts, which is significant drawback for B2MML and makes the idea of a "digital twin" difficult to implement. In order to support the idea of agile manufacturing with flexible MES services, authors focus on an implementation of ISA95 that is based on OPC UA communication middleware.

C. OPC UA-BASED COMMUNICATION MIDLEWARE FOR MES

1) OPC UA

OPC UA is an object-based and service-oriented communication middleware that not only supports the exchange of information but also organises the information models [26]. It is based on the client-server communication principle. Secure communication that uses *SecureChannel Service Set* is established by an OPC client. The stateful client-server connection is managed by the services that are collected in *Session Service Set*. The address space of the OPC UA server exposes both the data and the relevant information models that can be browsed by any connected client that uses the *View Service Set* for simple navigation through the server's address space or through the *Query Service Set* to perform complex browsing of the server's information content. Such an approach enables clients to have dynamic information access without any requirement for prior knowledge about the address space content. Next, clients can read or modify the available data using the *Attribute Service Set*. Clients can use subscription-based access to wait for the information about any data change or new information that is generated by using the *MonitoredItem Service Set*. Clients can also modify the server's address space with the support of the *NodeManagement Service Set*.

The information is organised in an object-oriented manner in which the structure and meaning of each data item are determined by the relevant type definitions. The basic OPC UA types are defined by the standard and are used to arrange the variables, objects, data and references that show the relationships between the information items [27]. The model can be easily expanded according to the requirements of the application by applying the inheritance mechanism. The OPC UA address space has a mesh structure and is composed of a number of *OPC Nodes*, which are connected by *OPC References*. Each node has to belong to one of eight classes, which determine its application context. The possible node classes are: Object, ObjectType, Variable, VariableType, DataType, Method, View and ReferenceType.

2) DIGITAL TWIN

The authors focus on a "*digital twin*" of the physical material items that exist in a manufacturing system that can be represented in the OPC UA address space by their an OPC UA Nodes. These nodes must belong to an OPC UA Object class (Fig. 6). The information structure is presented by the object's type definition, which is expressed by another OPC UA node that belongs to the ObjectType class. Both nodes are linked by a reference *HasTypeDefinition*. The references are used to describe any relationship between two OPC UA nodes. They can reflect an object's hierarchy (a reference it inherits

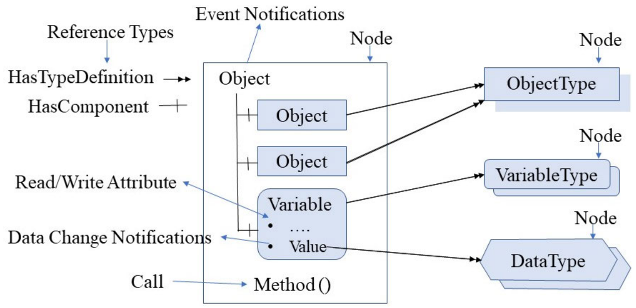

FIGURE 6. A digital tween for a material item that is created as an OPC UA object.

from the HierarchicelReferences type) or any other kind of relationship (such references are inherited from a NonHierarchicalReferences type such as the HasTypeDefinition). The *HasTypeDefinition* reference is one that is predefined by the OPC UA standard and is used to express the relationship between the type definition and the type instance. In addition to the standard reference types, custom types can be defined to specify information dependences that are relevant for a given application area.

The specific physical features of a material item are represented as *Variables*, which are also connected to a given OPC UA object by references. Each variable also has a type definition, which is indicated by a *HasTypeDefinition* reference. In a case in which a feature changes its value often during a production run, the object-variable relationship should be expressed by a *HasComponent* reference. In a case in which the features are more static, the *HasProperty* reference is used. In both cases, the *Value* attribute is used to represent the current state for a given physical feature. The data type that is available by a Value attribute is described by its *DataType* definition. There are a number of predefined OPC UA data types similar to those of references custom data types can be created.

3) COMMUNICATION MODELS

OPC UA Clients can read, write or subscribe for Value changes in accordance with the user permission policy that is specified for each OPC node. Information about any changes in the state of an object can be sent to clients as *Events* in accordance with the message exchange principle. Objects can offer clients the possibility to call *Methods* in order to change the object's state or behaviour. The Methods can utilise or return parameters. This mechanism unifies the way in which clients can interact with all instances of a given object type.

A view mechanism was introduced by OPC UA to simplify access to the server's address space. Clients can use views to decrease the number of information items that are visible. Only a subset of all of the nodes that is important for a given information model is presented. Nodes that belong to the *View* class are the entry points for clients that can use them as an entry node for browsing the server's address space. Views use references in order to expose a reduced subset of Nodes.

This mechanism can be used to represent different point of views in a material model by focusing on the selected material features that are important for a given manufacturing operation. Such an approach does not multiplicate the information about the materials by replicating data. Instead, it decreases the amount of redundant information that is not necessary, which will be invisible in a given view.

OPC UA supports access to the information that is stored in the address space *via* the read, write and subscribe mechanisms. The actual data access services that are available for OPC UA are grouped under the services: Data Access, Alarms and Events and Historical Data Access. Information exchange is begun by establishing a secure connection between the Client and the Server. Client-Server Sessions are defined over their underlying transport layers. The OPC UA defines the session-level services – Create Session, Activate Session and Close Session services. Such a structure robustly supports the interfaces in the event of communication errors because the sessions can survive any communication channel breaks.

The *Data Access* services include the classical client-server interaction that is implemented as the *Read* and *Write* services, which can interact directly with the selected properties of a given node. These mechanisms allow clients to directly access the information that is visible in the OPC UA address space. However, in the case of a Variable node and its Value attribute, more efficient communication can be established on the basis of the Publisher/Subscriber model. Such an approach is based on a subscription mechanism that is supported by the Subscription Service Set [28]. In this mode, clients receive new information only when Variable's value changes. The frequency used by server to check changes is individually set up by clients by using the *Sampling Interval* parameter. According to defined tracking conditions, an OPC UA Server sends new information to the client only if the information has changed. The subscription mechanism is also used in the case of information about *Events* and *Alarms*.

Data transfers are optimised by grouping data items into sessions for efficiency. The published request responses are not sent immediately but are queued by the server [29]. The responses (NotificationMessages) are sent back according to the Subscription's publishing interval. The NotificationMessages contain either Notifications of MonitoredItems (when there is an event to be reported to the client) or a KeepAlive message (a dummy message to notify the client that the server is operational).

The OPC *Historical Access* (HA) provides access to the historical archives that are used by applications such as the HMI(Human Machine Interface), Report Generation, Data Analysis etc. The data that is provided by the historical interface consists of historical records and, optionally, various calculated values such as the min, max, average etc. [30]. The HA enables the client to access historical variable values and events. It can read, write or annotate these data. The data can be located in a database, an archive or in another storage system. The OPC HA can be applied to the history of the materials and can be used to access the data repository. The advantage of such an access mode is its consistency and ease of management, which is difficult to obtain in systems that are based on copying data (distributed systems) or replicating data in local repositories.

Both Data Access and Historical Data Access follow a common memory model. Each client can freely choose the information in the server's address space and request access to the data in accordance with the requirements that they have defined. For events, the message exchange model was adopted. Other than the DA and HDA, clients do not subscribe to a specific event but select the information source, which is the OPC UA Object. Therefore, all subscribed clients receive precisely the same messages about events (which is not the case in the common memory model).

Based on the services described above, the OPC enables the complex data models that are adapted to the field in which they will be applied to be built. Nowadays, the OPC Foundation supports the different working groups that are involved in developing the industry-leading specifications, technologies, certifications and processes that are related to OPC UA. One of these is the ISA95 Working Group, which is responsible for defining and maintaining the OPC UA-based implementation for the ISA95 information models. The first release of the ISA95 OPC UA specification [31] includes support for models of Physical Assets, Equipment, Personnel and Materials. The authors focus on Material models. On this model, secure information access can be achieved from different IT systems including the production control level, Manufacturing Execution Systems (MES) and Enterprise Resource Planning (ERP) systems.

III. TELEMETRY DEVICE PRODUCTION – AUTOMATIC TEST LINE AND REWORK PRODUCTION STAND

A. PRODUCTION OF ELECTRONIC DEVICES

The production of electronic devices is performed as a series of successive production steps, which can be modelled as production segments according to the ISA95 standard. Each production segment is composed of a list of the materials that are consumed and produced, a list of the equipment that is used, a list of staff that are involved, a list of assets and a list of the parameters that are defined for a given segment. Most manufacturers of electronic devices divide the production process into two stages. The first stage is the frontend production, which begins with preparing the materials and usually ends at the stage in which an electronic circuit board is created. In most cases, the electronic components are installed using the SMD (Surface Mounted Devices) technology. Because of the complexity of this process and the large number of electronic components that are mounted on a PCB (Printed Circuit Board), the assembly of electronics mainly takes place on automated production lines, which is then followed by an automatic soldering stage.

The backend process includes the assembly of large elements or those that cannot be assembled using the

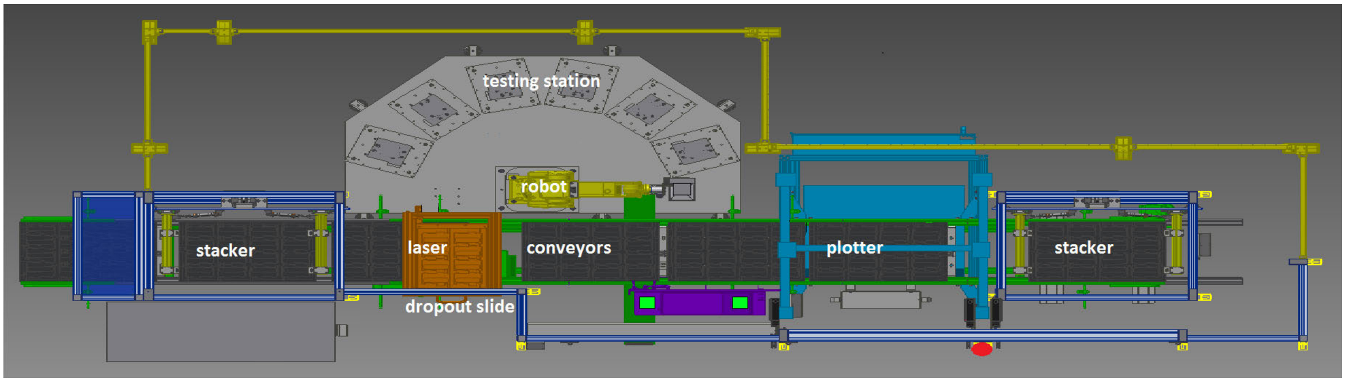

FIGURE 7. The layout of an automated test line.

SMD technique. The next step is the installation of additional elements such as connectors, displays, power supply, a housing etc. Between the individual production steps (or at the end of the stage), there may be tests whose task is to detect errors. Based on the error code, further decisions are made about the repair of the defective product or a decision is made to scrap the item. In a case in which the item is to be repaired, it is necessary to carry out the tests once again. Depending on the adopted production rules, the rework operation can only be repeated a limited number of times after which the product is qualified as scrap.

This architecture is also feasible for use in the production of GPRS (General Packet Radio Service) telemetry devices that is performed by the AIUT Company. These devices are mounted to the traditional gas meters as add-ons that are installed at the premises of an end-user that permit gas distribution companies to remotely access the usage metrics via a GSM (Global System for Mobile Communications). After the front-end production stage, a customised software is downloaded to the telemetry device, then the electrical and functional tests are performed. The operations above are fully automated. The layout of the automatic test line is presented in Fig. 7.

B. AN AUTOMATIC TEST LINE

An automatic test line consists of the following main components (from left to right): an input stacker that holds the palettes with the incoming telemetry devices, a manual supply drawer that is used as an additional source for feeding the line with devices, a total quality management integrated laser that is used for engraving product serial number on the device, six automated testing stations that are used for the electronic and optical testing of the devices, a dropout slide that is used to remove faulty devices from production, a plotter (resin injection device) that is used to flood the electronic boards with a moisture-protecting resin, an output stacker that holds the palettes with prepared, tested devices and a robot, which is a central element of the line.

The robot picks elements from the input palette or drawer and places them in one of the testing stations. The tests are performed in two stages: electrical and functional ones. In the first stage, electrodes are connected to the PCB and the wiring is checked. The second stage is for the functional tests in which the application software is loaded, and the response of the device is analysed for the given input signals from the tester. In the event that an error occurs at any of the stages, further tests are interrupted and information about the error code is recorded as the result of the operation. Devices for which there are errors are subject to repair.

Depending on the test result, the device is either removed from the production line via a drop-out slide or is moved to the laser station at which the serial number is engraved. Afterwards, the device is put into the output drawer. Full drawers are transported to the next station where the devices are filled with a moisture-protecting resin and are finally moved to an output stacker. In the case of devices that are removed from the line using the dropout slide, the whole process of their assembly and testing can either be repeated or continued at a rework station.

Palettes are transported between the stations for input into the output stacker is performed by a conveyor belt. Rework production stands are located outside of the test line and are used for manual repairs and backup operations for the devices that fail the tests on the automatic test lines. Repair operations can be connected with both hardware and software errors and can range from the verification and replacement of defective components, e.g. display, GSM slot or radio modules to the manual reload and testing of a device's firmware.

All of the devices that are produced that fail the functional or electrical tests on one of backend production lines are marked as defective and are scheduled for rework operations. The result of all of the tests along with the specific statuses of the devices and received error codes is part of the material record that is a "digital twin" for all of the devices or their components that are produced. The records are consistent with the ISA95 standard and are accessible by the OPC UA server.

C. THE REWORK STATION

When the rework station is activated, it appears in the system and enters an initialisation process. Firstly, the OPC server on which all of the products and production orders are gathered synchronises with the TQM system. Then, an XML with the values and a description of the data structures is downloaded and the rework database is updated. As the first initialisation step, the list of device types that are currently on the production line is downloaded.

The devices that require repair operations are transported to the rework station and their repair process can then begin. The order of the rework operations that are performed at a rework station is as follows: defective devices from the backend production lines are delivered to the station and stored near the repair and assembly zone.

A system for monitoring and controlling the production updates, production data and production orders provides an updated list of to-do devices, which is delivered to the rework personnel. Then, the selection of the device is possible using either an icon of the device or by typing the serial number of device's PCB. At this stage, the list of devices can be categorised or filtered according to the malfunction or assembly defect that was detected. When the device is selected, a repair instruction is displayed in relation to the provided error code and the rework operations can be performed. An example of an instruction is presented in Fig.8.

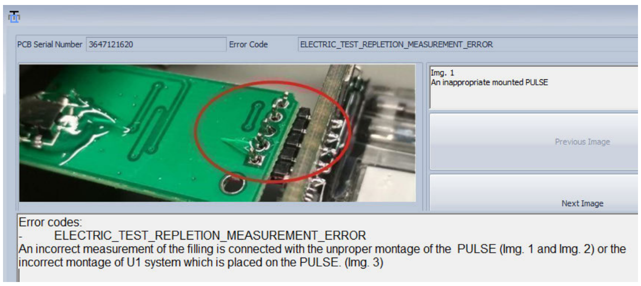

FIGURE 8. Display of a repair instruction.

After the successful repair, it is necessary to indicate which repair operations were performed, either by selecting the methods that already exist or by inserting new ones. After this, the updated production and product information are moved to the archive of the rework database and they disappear from the list. In addition, the status of the device is updated in the production system database so it can continue the production process. The list of all of the rework operations is stored in the database so it can be browsed for details, e.g. the operator who performed the repair, the error codes of the defects, the solutions used, or the time required for the repairs.

In the event that the rework operations cannot be performed at specific moment, the repair can be postponed, or the device can be scrapped. In such case, a remark can be included in the description as a comment and then saved in the archive. When the rework operations are continued for such a postponed device, the inserted comment is displayed. The rare example in which there is no possibility to repair the specific device ends with the device being scrapped or disassembled and its production order being removed from the system.

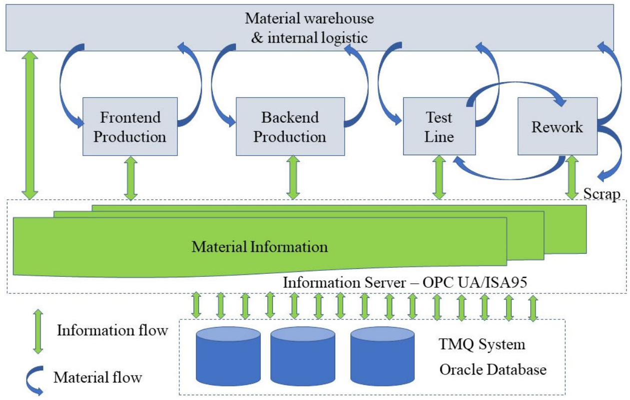

FIGURE 9. The architectural model for managing the material information.

IV. USE CASE STUDY: IMPLEMENTATION OF A MATERIAL MODEL WITH A FOCUS ON REWORK OPERATION

A. MATERIAL INFORMATION

The authors developed a material model that is used to support rework operations. The model is also used by a number of the MES services that are dedicated for supporting short-run production. The analysis below refers both to the material ontology description defined under ISA95 and the modelling rules that are available for an OPC UA address space.

Information about the materials that are required by services described above can be supported in a static manner using B2MML files or in a dynamic manner using OPC UA support for ISA95. B2MML-based communication is typical in the case of data exchange between ERP and MES. Such an approach is not convenient to gain selective access to the required material properties during rework operations because the information changes during the production steps that are performed.

In the proposed approach, the authors use OPC UA to model material information. The ISA95 properties are represented as OPC UA variables and the ISA95 classes are modelled as OPC UA objects. This approach is in line with the guidelines given by the OPC MES working group [31]. Therefore, OPC UA clients can selectively access the required properties of a material via the subscription mechanism, receive the historical data or subscribe to an event-based communication. An OPC UA server is not a source of information but only acts as the communication interface that provides access to the material information under the ISA95 material models. The ISA95 models are adjusted to the actual products and the information content is shared according to the given context of use. The generic architectural model for managing the material information that is used by Aiut Ltd. for the production of telemetry devices is shown in Fig. 9.

The material information (green arrows) follows a physical material flow (blue arrows) for all of the production steps. Materials are consumed, produced and processed during the front-end and back-end production steps. The telemetric devices are unambiguously indicated by their serial numbers, which are printed on PCBs (Printed Circuits Boards). Each device is then tested by an Automatic test line (see previous section). If all of the electrical and functional tests are successful, a unique serial number for the device that is different than the PCB number is generated by the TMQ, which is used in the production management system and is a part of the ERP system. This number identifies the final product and binds it with a customer's order, and therefore, it is engraved on the telemetric device by a laser printer. In the case of any error that is detected, the produced device is sent to a rework station in order to repair the problem. Accordingly, the Material Information is supplied with data about the detected errors. If the problem is solved, the device is retested, and the new test results are added to the Material Information. Otherwise, when the error is not able to be repaired, the device is scrapped. All of the production processes are supervised by a TMQ that utilises an Oracle database.

B. THE MATERIAL MODELS

The material models were prepared according to the ISA95 rules that were described in chapter II.B. ISA95 defines four levels of information models as is illustrated in Fig. 4: a MaterialClass that contains a generic description for the material features that are common between the class members, a MaterialDefinition that is dedicated to describing the specific kind of material, a MaterialLot that is related to a physical batch of materials and a MaterialSublot that can be used for a detailed description of every physical item of material. In the presented case, the MaterialClass example is named TelemetryDevice_Type and is used to describe all of the electronic devices that are produced by Aiut's TMQ. The MaterialDefinition example is APULSE, which describes one type of the smart gas meters that are produced by Aiut. Both models are defined and utilised by the production technology management services. In order to collect the production data services, the MaterialLot – APULSE_LOT and MaterialSubLot – APULSE_SUBLOT models are defined and used. respectively.

The detailed material features are described by all four models as being material properties: MaterialClassProperty for the MaterialClass models, MaterialDefinitionProperty for the MaterialDefinition and MaterialLotProperty for the MaterialLot and MaterialSubLot models. The property definitions can be nested in the tree schemas in order to express complex models. In the case of APULSE, the property examples are: SerialNo - serial number of each device, Status – device status during production, ERROR – error code, or more detailed information such as, for example, a Type property that is composed of information about: the CRYPTOGRAPHY_KEY_TYPE, MAC_KEY_TYPE, IC_DEVICE_FIRMWARE_VERSION and others. According to the ISA95 rules, more detailed properties are mapped to the relevant part of the more generic models. An example of an ERROR_CODE property defined for TelemetryDevice_Type and APULSE material schemas is shown in Fig. 10.

Although each property is defined by its unique ID, more information that describes the property features can be found in the property's Description attributes, which form a list that can express the different aspects that are related to a given material feature. Similarly, the Value of a property is also expressed by a list of its attributes. In the presented example, the Description of the ERROR_CODE in MaterialClass model gives only very generic information about a property, which contains the error code for the device. In the case of the MaterialDefinition model, the same property has three additional Description attributes that define the SQL query templates that are used to access the information in the TMQ System. These enable the actual SQL queries that will be used for the MaterialSubLot model to be built. All of the information models described above are stored as B2MML files in the TMQ System. They are used to provide a detailed description for the production technology of a given model of electronic gas meter, including information that is specific to different software and hardware versions. The MaterialDefinition models form the technological patterns for production that are shown in Fig. 9 as green arrows directed from bottom $->$ top. Similarly, the MaterialSubLot models are used as templates for the information that is collected while tracking the flow of a material and are depicted in Fig.9 as top $->$ bottom arrows.

Since the information that is represented by B2MML cannot be used for on-line processing by production systems, the relevant OPC UA information models that are compatible with the ISA95 schemas (see section II.C) were created. The models and communication interfaces are used together to support the material information exchange that has to be adjusted according to the required context of its use and should reflect the features (ISA95 properties) of a given product type. In order to ensure flexibility, the information model is not embedded or hardcoded in any part of the system. All of the necessary information about the material structure is generated dynamically in the OPC UA address space based on a material information structure that is supported by relevant B2MML files.

C. OPC UA ADDRESS SPACE

The object-oriented definitions for the material types are based on the Material Definition in ISA95 and are visible in the address space of the OPC UA Server. These definitions are used during the creation of the Material Information objects as reference-type information. The structure of the information is adjusted to the physical features of each produced device by a type selection. The B2MML files are used to exchange information about material data structure that is based on the ISA95 Material Class and Material Definition. The owner of the production technology models is a TMQ system. The OPC UA server provides an adjusted context-based information about part of this model.

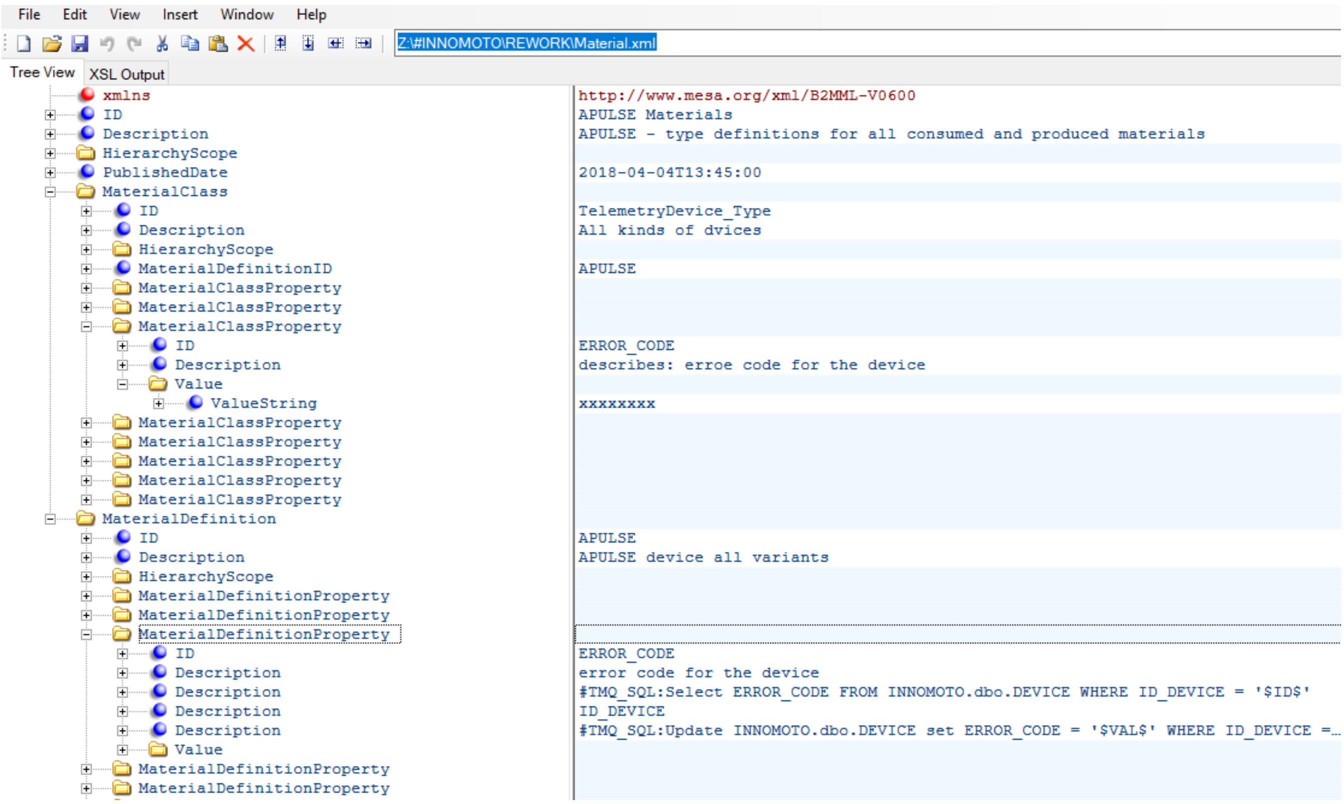

FIGURE 10. **An example of the ERROR_CODE property in the B2MML schema that was created under the ISA95 models: materialClass and materialDefinition.**

The OPC Foundation prepared an implementation guide for the implementation of ISA95 [25], which describes the information models that are defined in the core components technical specification (CCTS) that was developed by the ebXML project, which was organised by UN/CEFACT and ISO TC 154. This also includes the supplementary components that provide metadata that refine the value domain. This standard defines a data representation model that is compatible with the CCTS. The value of a ???content ???is represented by the Value Attribute of a Variable, and the supplemental components that are defined in the CCTS are represented as the Properties of the DataType. In addition, these supplemental Properties can optionally be defined for the Variables. However, for some DataTypes or VariableTypes, the supplemental Properties could sometimes have different values than the default value. A client should try to read the supplemental Property from the Variable first. If the Property does not exist on a variable then the Client should read the supplemental Property from the CDT DataType. This model was used for OPC UA-based Information server that was designed for the production of telemetry devices.

The CCTS specification defines several predefined object-based type definitions. In the case of the material model, five object types that inherit basic OPC UA ObjectType are defined: MaterialClassType, MaterialDefinitionType, MaterialLotType, MaterialSubLotType and MaterialTestSpecificationType. These directly refer to the ISA 95 models that are shown in Fig. 4. The first two types are used to structure the technological information about the material class and the specific material definition, the next two are used as a template for information about materials that are consumed and produced, which focus on a specific lot and sublot of material, respectively. The actual properties of a material are expressed by variables that belong to one of four classes that are based on the OPC UA VariableType: MaterialClassPropertyType, MaterialDefinitionPropertyType, MateriaqlLotPropertyType (also used for information about the sublot) and MaterialTestResultType. The information structure is presented by objects, while the information content is used by the variables. All of the nodes in OPC UA models are linked by references. The CCTS introduces two new reference types that are based on a basic OPC UA reference HasComponent. The HasISA95ClassProperty references are used to model the technological information and the HasISA95Property references are used to model the actual material information. The use case study presented in this paper also follows these ISA 95 types. The relevant models were created for both the technological design and for the data about actual electronic devices being produced. The example part of the OPC UA address space that was used for the APULSE material definition is shown in Fig. 11.

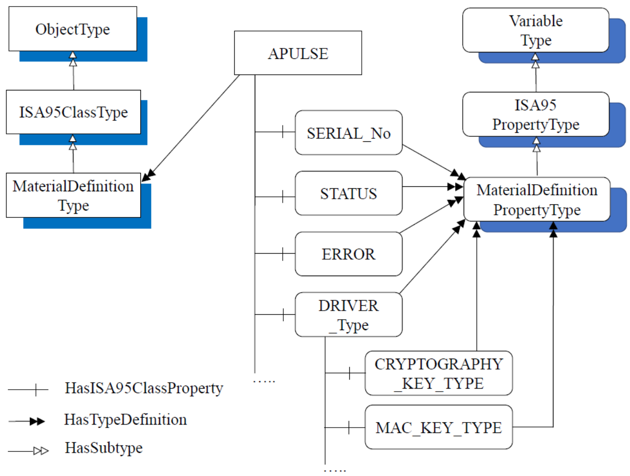

FIGURE 11. **APULSE material definition in the OPC UA address space.**

D. THE OPC UA AND ISA95 MODEL FUSION

The MaterialLot and MaterialSublot information is generated dynamically based on the descriptions that are given by TMQ system in the B2MML files. This mechanism utilises the ISA 95 Description field, which is a component of the MaterialDefinitionProperty object. The Description fields are used to exchange the templates for SQL queries that are necessary for bidirectional communication between the OPC UA server and the TMQ. The Query templates are defined by TMQ (the database owner) and reflect the structure of the SQL database (tables, columns, relations), which is encapsulated in the B2MML file and is available to the OPC UA server. Based on the B2MML descriptions, the relevant OPC UA types are created and the information about SQL queries is passed to OPC UA server. The example part of OPC UA address space that contains SQL queries for the error code value is shown in Fig. 12.

Since ISA95 does not limit the number of instances of the Description field (0…infinity) for a given MaterialDefinitionProperty, the Description fields can be used to define all of the necessary templates for the SQL queries and can also be used for any descriptive approach that is required by the ISA95 modelling rules. The #TMQ_SQL:Select and #TMQ_SQL:Update keywords were introduced in the proposed system as templates for the Select and Insert queries. The actual SQL queries are generated based on these when the actual instances of a MaterialLot or MaterialSubLot are created. This is performed by joining the SQL templates with the ID of the material and therefore valid SQL queries can be used to communicate with the TMQ database. In a case in which such a communication with TMQ is not necessary (for example for properties with constant values or for temporary information), the SQL templates are not placed in Description field.

When a new material item is prepared for production, the TMQ gives this information to the OPC Server by updating the MaterialsUnderProduction list. The OPC UA server creates a new instance of the material in its address space using the material ID and the relevant material type definition (object-based material types are created based on the B2MML content). From this point, any OPC UA client can use or update the information about a material instance that is indicated by a given ID. In order to follow the information changes, OPC UA clients use a subscription mechanism that is supported by the OPC UA Data Access. In the event of any change in the value of a given material property, the OPC subscription mechanism ensures that all of the subscribed clients have up-to-date information.

The OPC UA Data Access subscription mechanism provides the most recent (more up to date) values for the selected material properties. The subscription does not support any information about the history of the changes that in some cases may be important. For example, information about the error code should not only reflect the last error status but also the entire history of the errors that have been detected for a given material item. In such a case, the OPC Historical Data Access is used. As in the case of Data Access, the OPC UA Server only acts as a communication interface for the Historical Data Access and is not responsible for storing information. The actual historical data are stored in an Oracle database that is managed by the TMQ. The relevant templates for SQL queries are exchanged between the TMQ and OPC UA server via B2MML files. Only some of the material properties require historical data support. For example, the ID number does not change during the entire production process and therefore its history is not supported. In order to mark Material Properties with historical data access an OPC UA Historising attribute for relevant data item is set to true. Both the Data Access and Historical Data Access modes are selective under the OPC UA Client's demands. Another communication approach is used in the case of OPC UA Events.

Event information is sent to all Clients that have subscribed to them using event source addressing and a filtering mechanism. The source of events can be a single material item (Material SubLot), a whole batch of material (Material Lot), all of the devices of given type under production or all of the Telemetry devices. The event message is merged with information about the current error status, which is also supported by Data Access communication services and by Historical Data Access mechanism. Event information is transmitted via a message exchange so that no piece of information can be lost. For example, the event "Rework finished" is used by TQM scheduler that is connected to rework station by OPC UA client. Therefore, information about the completed rework is used to plan new tasks for the Automatic Test Line or for adding new telemetry devices to the production plan.

The proposed information models directly reflect the RAMI4.0 approach. The current state of a material, its history and event information are consistent and is directly supported by instances of the Material Information objects that are available to the relevant OPC UA services. Therefore, OPC UA supports the "digital twin" of an actual material item. In the

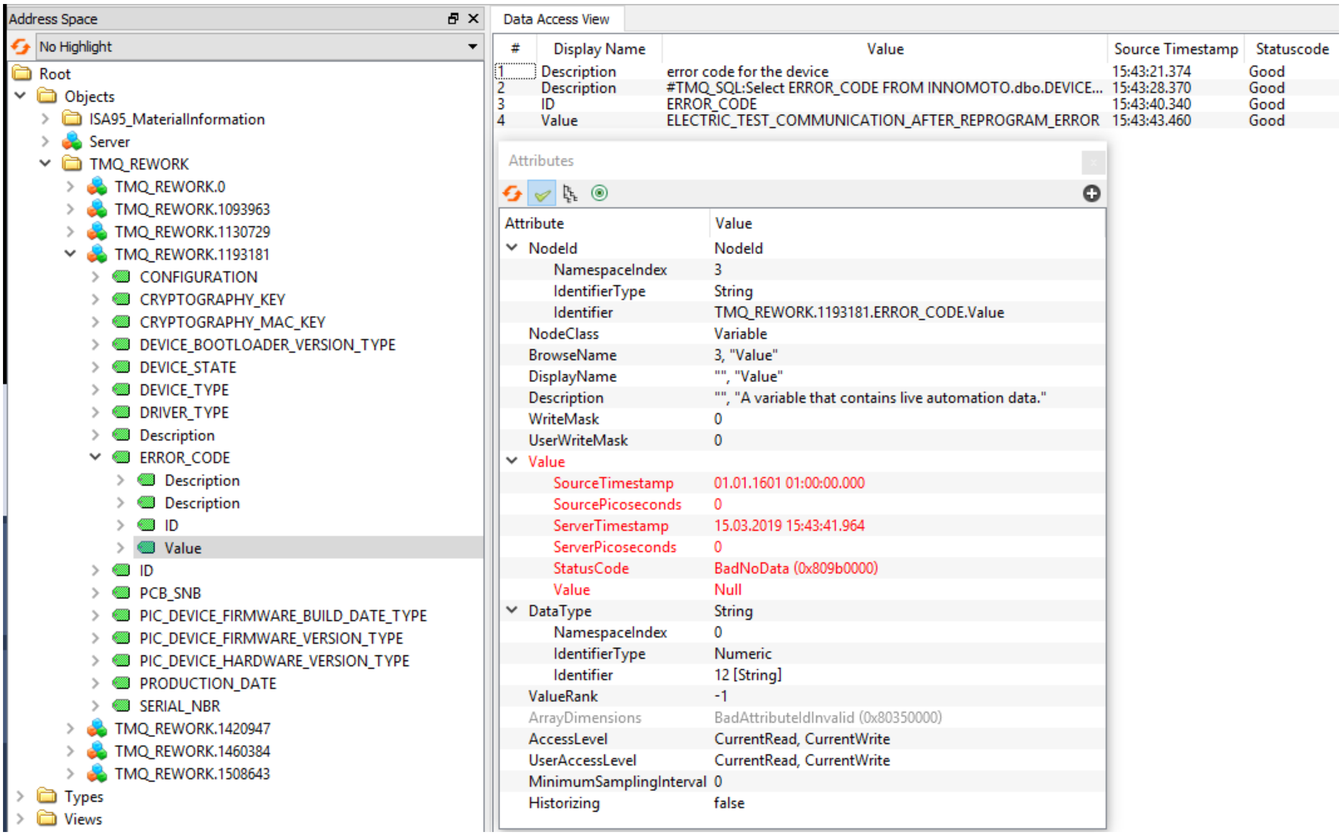

FIGURE 12. **Address space of OPC UA server with SQL queries for ERROR_CODE property.**

automation pyramid approach, such a set of information was composed of many production systems but not by the material itself. This created a heterogeneous access to data that could cause information inconsistency. This drawback has been eliminated in the proposed approach in which the material exposes all of the necessary data regardless of the underlying production process. During the subsequent production steps, the information about the material is enriched but the material model remains consistent. Such a material-centric approach makes production more flexible and the material model can remain unchanged even in the event that the production technology has to be changed.

E. ISA95 EXTENSIONS

Other aspects of RAMI4.0 that are not directly supported by ISA95 models are the dependences between the different types of materials (in the presented case telemetric devices) and their different production versions. Electronic devices are produced in many variants and their hierarchy can be represented in the form of a tree in which the root represents the features that are common to the entire family of products and in which the subsequent branches provide the parameters that are narrower and are relevant to the specific technical details. The proposed approach uses a representation by the inheritance mechanism, which is visible in the OPC UA address space.

Another issue connected with material modelling according to the RAMI4.0 approach under ISA95 is the presentation of the evolution of a material between the successive versions of the same product. This is a typical situation when the next version of a telemetry device differs from the previous by a new hardware design, new software or new adapters that were unavailable in the previous product version. RAMI4.0 requires that product development should be reflected in the relevant information models. Unfortunately, the meta information that has been defined by ISA95 is composed of the two models: a Material Class that can be used to describe the generic properties of a material class and the Material Definition that defines very specific patterns for a given material type. Both classes can be used to describe the properties of the materials that are used for the production or for the finished products. However, they do not permit the relationship between the members of product family and the development of subsequent versions of the material to be described as is required by the Vale Stream axis of the RAMI4.0 model (see Fig. 1). It is also not sufficient to reflect changes in material description. Subsequent versions may require additional properties that were not present in the previous versions, modifications of a selected property or the removal of properties that have become unnecessary as a result of product development.

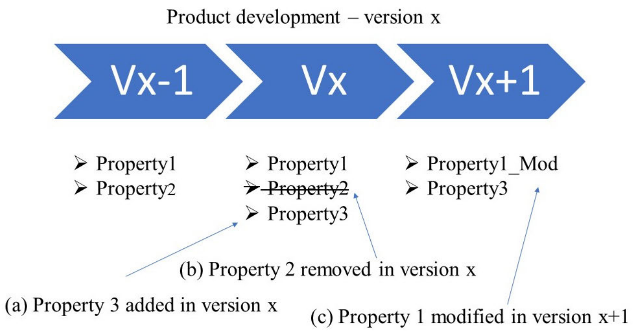

FIGURE 13. **Product development - changes in a material description.**

This problem is illustrated in Fig. 13 in which three typical changes in a material description are presented: (a) a new material feature should be modelled as a new property, (b) an existing material feature was changed so the relevant property in the model should also be changed or (c) the material feature is no longer present or no longer important for the model so the relevant property should be removed from the model. Although the addition of a new property is supported by OPC UA and may be solved by using the type inheritance mechanism, the removal of unnecessary properties or their modification is not supported. The authors applied a solution that is compliant with both the ISA95 and OPC UA standards that enables the changes that are related to product development and their modelling to be presented. The solution is based on additional information that is stored in the Value field of the MaterialDefinitionPropertyType, which can be expressed as a list according to the relevant B2MML definition (like in the previously presented mechanism for an SQL template passed in the Description field).

The Value field in ISA95's MaterialDefinitionPropertyType is intended to determine the range of values for a given property. It can be also used to give examples that can be used for actual objects. The authors additionally use it to develop the material model. In the presented solution, the first two instances of Value field are used according to the assumptions as defined by the ISA95 type definition: the first is used to define the default value for a property while the second Value occurrence is used for its type definition. The type definition is used to automatically generate the OPC UA types that are based on the ISA95 schemas. The next two of items on the Value are used to describe the evolution of the model. The keyword #active means that the property is used in the current version of the description. In such a case, the relevant variable on an OPC UA server will be generated, while the keyword #inactive means that the property was removed and is no longer used. In such a case, the property will not be initiated in the OPC UA object. The information about the unused property is left to follow the model evolution. The second instance for a Value that is preceded by #VersionNb indicates the material version when the change was introduced.

Since all of the models are identified by their versions, it is possible to reconstruct the evolution of the material model and to find the relevant models for the previously used versions of the material description. In the case of b, two operations have to be applied on the model to change a given property. The old value is removed by the #inactive tag and the new one with #active is added (both with the number of the material version in which the change was introduced). The proposed solution is compatible with OPC UA because only the active properties are visible and with ISA95 in which the inactive properties can be used to track the evolution of the model. The previous versions of the material can be generated using the proposed history of its evolution, which is contained in the B2MML files. The information about the material version together with information about any version changes is used to generate the dynamic OPC UA material type.

V. CONCLUSION

This paper presents the results of research that is related to information modelling dedicated for the new generation of manufacturing systems. It focuses on the application of the ISA95 and OPC UA standards that are required to create a "digital twin" of the materials that are used in the production of highly customised electronic devices. The main advantage of the proposed solution is the replacement of the information modelling from the centric-based production system with a product flow driven information system. Material Information becomes the "digital twin" of the actual (physical) material during the entire production process. Changes in the production technology do not force changes in the MES services that use the material information according to the proposed polymorphic model. In the event that the production technology change is important for the model, the material features are adjusted in the model by the relevant properties. Changes can be traced under the solution that was proposed for modeling the evolution of a material.

Although, the proposed approach is based on the ISA95 models that are implemented by communication middleware that is based on the OPC UA standard, some extensions were necessary to fit to the RAMI 4.0 approach. The proposed solution minimises the new elements to make it compatible with the other systems that are used by industry. The main extensions are the additional use of the Description field (SQL templates stored in for the PropertyType in the Material Class) and support for developing an information model by keeping the unused or changed material properties that was modelled by the Value attribute (history of property changes by the fields in the Material Class). The proposed solution enriches the static ISA95 models by the dynamic data presentation that is performed by OPC UA servers. The communication features of OPC UA that are used are: Data Access for the selective and continuous support of the information about the material properties that are required by individual OPC UA clients, Historical Data Access to create a consistent model that binds the actual values of selected material parameters with their historical data, which are owned and stored by ERP systems, an Event mechanism to automatically follow all of the important changes in the production status that have to be traced by Manufacturing Execution Systems.

The authors believe that the proposed approach can easily be extended to fulfil other material models in order to follow the new requirements, which have resulted from the idea of a "digital twin" and the models that are defined by RAMI4.0.

REFERENCES

[1] B. Maskell, "The age of agile manufacturing," *Supply Chain Manage.*, vol. 6, no. 1, pp. 5–11, 2001, doi: 10.1108/13598540110380868.

[2] J. Wan, M. Yi, D. Li, C. Zhang, S. Wang, and K. Zhou, "Mobile services for customization manufacturing systems: An example of industry 4.0," *IEEE Access*, vol. 4, pp. 8977–8986, 2016, doi: 10.1109/ACCESS.2016.2631152.

[3] R. Cupek, A. Ziebinski, M. Drewniak, and M. Fojcik, "Knowledge integration via the fusion of the data models used in automotive production systems," *Enterprise Inf. Syst.*, vol. 13, nos. 7–8, pp. 1094–1119, Jun. 2018. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/17517575.2018.1489563

[4] F. Tao and M. Zhang, "Digital twin shop-floor: A new shop-floor paradigm towards smart manufacturing," *IEEE Access*, vol. 5, pp. 20418–20427, 2017, doi: 10.1109/ACCESS.2017.2756069.

[5] D. M. Tilbury, "Cyber-physical manufacturing systems," *Annu. Rev. Control, Robot., Auton. Syst.*, vol. 2, pp. 427–443, May 2019, doi: 10.1146/annurev-control-053018-023652.

[6] T. Sauter, "The continuing evolution of integration in manufacturing automation," *IEEE Ind. Electron. Mag.*, vol. 1, no. 1, pp. 10–19, May 2007, doi: 10.1109/MIE.2007.357183.

[7] Y. Lu and X. Xu, "Resource virtualization: A core technology for developing cyber-physical production systems," *J. Manuf. Syst.*, vol. 47, pp. 128–140, Apr. 2018, doi: 10.1016/j.jmsy.2018.05.003.

[8] K. Ding, F. T. S. Chan, X. Zhang, G. Zhou, and F. Zhang, "Defining a digital twin-based cyber-physical production system for autonomous manufacturing in smart shop floors," *Int. J. Prod. Res.*, vol. 57, no. 20, pp. 6315–6334, Jan. 2019, doi: 10.1080/00207543.2019.1566661.

[9] G. Engel, T. Greiner, and S. Seifert, "Ontology-assisted engineering of cyber–physical production systems in the field of process technology," *IEEE Trans. Ind. Informat.*, vol. 14, no. 6, pp. 2792–2802, Jun. 2018, doi: 10.1109/TII.2018.2805320.

[10] K. Ding, X. Zhang, F. T. S. Chan, C.-Y. Chan, and C. Wang, "Training a hidden Markov model-based knowledge model for autonomous manufacturing resources allocation in smart shop floors," *IEEE Access*, vol. 7, pp. 47366–47378, 2019, doi: 10.1109/ACCESS.2019.2909306.

[11] R. Cupek, H. Erdogan, L. Huczala, U. Wozar, and A. Ziebinski, "Agent based quality management in lean manufacturing," in *Computational Collective Intelligence* (Lecture Notes in Computer Science), vol. 9329. Cham, Switzerland: Springer, 2015, pp. 89–100, doi: 10.1007/978-3-319-24069-5_9.

[12] K. Schweichhart, "Reference architectural model industrie 4.0 (RAMI 4.0)," in *Proc. Standardization Reference Archit., Plattform Ind. 4.0 Publikationen Plattform Ind. 4.0 Stand*, Apr. 2016, pp. 1–15. [Online]. Available: https://ec.europa.eu/futurium/en/system/files/ged/a2schweichhartreference_architectural_model_industrie_4.0_rami_4.0.pdf

[13] T. Sauter, "Integration aspects in automation—A technology survey," in *Proc. ETFA*, Catania, Italy, vol. 1, Sep. 2005, p. 9, doi: 10.1109/ETFA.2005.1612688.

[14] H. S. Kang, J. Y. Lee, S. Choi, H. Kim, J. H. Park, J. Y. Son, and S. D. Noh, "Smart manufacturing: Past research, present findings, and future directions," *Int. J. Precis. Eng. Manuf.-Green Technol.*, vol. 3, no. 1, pp. 111–128, Jan. 2016, doi: 10.1007/s40684-016-0015-5.

[15] R. Cupek, A. Ziebinski, L. Huczala, and H. Erdogan, "Agent-based manufacturing execution systems for short-series production scheduling," *Comput. Ind.*, vol. 82, pp. 245–258, Oct. 2016, doi: 10.1016/j.compind.2016.07.009.

[16] B. Chen, J. Wan, L. Shu, P. Li, M. Mukherjee, and B. Yin, "Smart factory of industry 4.0: Key technologies, application case, and challenges," *IEEE Access*, vol. 6, pp. 6505–6519, 2017, doi: 10.1109/ACCESS.2017.2783682.

[17] J. Harrington, *Computer Integrated Manufacturing*. Melbourne, FL, USA: Krieger, 1979.

[18] K. Schmidt and L. Bannon, "Taking CSCW seriously," *Comput. Supported Cooperat. Work*, vol. 1, nos. 1–2, pp. 7–40, Mar. 1992, doi: 10.1007/BF00752449.

[19] F.-T. Cheng, E. Shen, J.-Y. Deng, and K. Nguyen, "Development of a system framework for the computer-integrated manufacturing execution system: A distributed object-oriented approach," *Int. J. Comput. Integr. Manuf.*, vol. 12, no. 5, pp. 384–402, Nov. 2010, doi: 10.1080/095119299130137.

[20] M. McClellan, *Applying Manufacturing Execution Systems*. Boca Raton, FL, USA: CRC Press, 1997.

[21] C.-Y. Huang, "Distributed manufacturing execution systems: A workflow perspective," *J. Intell. Manuf.*, vol. 13, no. 6, pp. 485–497, Dec. 2002, doi: 10.1023/A:1021097912698.

[22] B. K. Choi and B. H. Kim, "MES (manufacturing execution system) architecture for FMS compatible to ERP (enterprise planning system)," *Int. J. Comput. Integr. Manuf.*, vol. 15, no. 3, pp. 274–284, Nov. 2010, doi: 10.1080/09511920110059106.

[23] T. J. Williams, P. Bernus, J. Brosvic, D. Chen, G. Doumeingts, L. Nemes, J. L. Nevins, B. Vallespir, J. Vlietstra, and D. Zoetekouw, "Architectures for integrating manufacturing activities and enterprises," *Comput. Ind.*, vol. 24, nos. 2–3, pp. 111–139, Sep. 1994, doi: 10.1016/0166-3615(94)90016-7.

[24] B. Scholten, "History of development of ISA95," in *The Road to Integration: A Guide to Applying the ISA-95 Standard in Manufacturing*. Durham, NC, USA: ISA, 2007, pp. 26–27.

[25] *Business To Manufacturing Markup Language Operations Definition Version 6.0*, B2MML-OperationsDefinition, Mesa, AZ, USA, Mar. 2013.

[26] J. Lange, F. Iwanitz, and T. J. Burke, *OPC—From Data Access to Unified Architecture*. Berlin, Germany: VDE Verlag, 2010, pp. 111–130.

[27] R. Cupek, K. Folkert, M. Fojcik, T. Klopot, and G. Polaków, "Performance evaluation of redundant OPC UA architecture for process control," *Trans. Inst. Meas. Control*, vol. 39, no. 3, pp. 334–343, Mar. 2017, doi: 10.1177/0142331215603792.

[28] W. Mahnke, S. H. Leitner, and M. Damm, *OPC Unified Architecture*. New York, NY, USA: Springer-Verlag, 2009, pp. 156–175.

[29] X. Hong and W. Jianhua, "Using standard components in automation industry: A study on OPC specification," *Comput. Standards Interfaces*, vol. 28, no. 4, pp. 386–395, Apr. 2006, doi: 10.1016/j.csi.2005.05.001.

[30] M. Bochenek, M. Fojcik, and R. Cupek, "OPC historical data access—OPC foundation toolkit improvement suggestions," in *Proc. Int. Conf. Comput. Netw.*, Ustron, Poland, Jun. 2011, pp. 338–347, doi: 10.1007/978-3-642-21771-5_37.

[31] *OPC Unified Architecture for ISA-95 Common Object Model Companion Specification Release 1.00*, OPC Found., Scottsdale, AZ, USA, 2013.

[32] *ANSI/ISA-95.00.03-2005 Enterprise-Control System Integration, Part 3*. Accessed: Jan. 12, 2015. [Online]. Available: https://isa-95.com/ansiisa-95-00-03-2005-enterprise-control-system-integration-part-3-downloadable

RAFAŁ CUPEK received the M.Sc., Ph.D., and D.Sc. degrees in computer science from the Silesian University of Technology, in 1991, 1998, and 2019, respectively. His qualifications are computer networks and distributed systems, industrial real-time systems, industrial process visualisation, and manufacturing execution systems. He focuses on a new generation of computer systems used by industry, including machine-to-machine communication (M2M), information models for digital twins, and effective and robust communication. He works both on science and research, and as a Lecturer at the Silesian University of Technology. He took part in numerous research project on EU and national level.

MAREK DREWNIAK received the Ph.D. degree from the Silesian University of Technology. He is currently the Head of the Research and Development Department, AIUT Ltd. He professionally bound with automation control and industrial informatics, formerly, he was implementing automation control solutions for industry. He is currently responsible for the development of products and management of research programmes. He was a participant of several research and development projects and currently involved in works related to energy efficiency analyses of production stations and mobile robotics.

ADAM ZIEBINSKI received the M.Sc., Ph.D., and D.Sc. degrees in computer science from the Silesian University of Technology, 1996, 2002, and 2019, respectively. He has been with the Institute of Informatics, Division of Informatics Devices, Faculty of Automatics, Electronics, and Computer Science of the Silesian University of Technology, Poland, since 1996. He is author or coauthor of several dozen scientific publications and three books and holds three patents and six patent applications. His research interests include computer architecture, embedded systems, hardware description language, programmable devices, and programmable logic controllers. He has experience in both academic and industrial research. He participated in research projects at the EU and national level. He is a Supervisor of Academic Student Association - Industrum.

MARCIN FOJCIK received the M.Sc. degree, in 1991. In 2006, he was a Førstelektor. He was with the Silesian University of Technology, from 1990 to 2002, and with the Sogn og Fjordane University College, from 2005 to 2017. He has been with the Western Norwegian University of Applied Sciences, since 2017. His main expertise is in information and communication technology, network communication, and embedded systems. His research involves the evaluation and application of existing protocols in different areas, such as industry, telecare, medicine, and smart home or autonomous vehicle.