Concept and Implementation of a Software Architecture for Unifying Data Transfer in Automated Production Systems

Utilization of Industrie 4.0 Technologies for Simplifying Data Access

Emanuel Trunzer[0000-0002-4319-9801], Simon Lötzerich and Birgit Vogel-Heuser[0000-0003-2785-8819]

Institute of Automation and Information Systems
Technical University of Munich, Munich, Germany
{emanuel.trunzer, simon.loetzerich, vogel-heuser}@tum.de

Abstract. The integration of smart devices into the production process results in the emergence of cyber-physical production systems (CPPSs) that are a key part of Industrie 4.0. Various sensors, actuators, Programmable Logic Controllers (PLCs), Manufacturing Execution Systems (MES) and Enterprise Resource Planning (ERP) systems produce huge amounts of data and meta data that can hardly be handled by conventional analytic methods. The main goal of this work is to develop an innovative architecture for handling big data from various heterogeneous sources within an automated production system (aPS). Moreover, enabling data analysis to gain a better understanding of the whole process, spotting possible defects in advance and increasing the overall equipment effectiveness (OEE), is in focus. This new architecture vertically connects the production lines to the analysts by using a generic data format for dealing with various types of data. The presented model is applied prototypically to a lab-scale production unit. Based on a message broker, the presented prototype is able to process messages from different sources, using e.g. OPC UA and MQTT protocols, storing them in a database and providing them for live-analysis. Furthermore, data can be anonymized, depending on granted access rights, and can be provided to external analyzers. The prototypical implementation of the architecture is able to operate in a heterogeneous environment supporting many platforms. The prototype is stress tested with different workloads showing hardly any response in the form of longer delivery times. Thus, feasibility of the architecture and its suitability for industrial, near real-time applications can be shown on a lab-scale.

Keywords: Automated Production System (aPS), Big Data Applications, Cyber-physical Systems (CPS), Data Acquisition, Data Analysis, Heterogeneous Networks, Industrie 4.0, Industry 4.0, Internet of Things (IoT), Message-oriented Middleware, Systems Architecture

O. Niggemann and P. Schüller (Eds.), *IMPROVE – Innovative Modelling Approaches for Production Systems to Raise Validatable Efficiency*, Technologien für die intelligente Automation 8, https://doi.org/10.1007/978-3-662-57805-6_1

1 Introduction and Motivation

Globalization and high competitive pressure require manufacturing companies to develop new solutions such as the digitalization of existing production processes, massive information exchange, and development of new business models. Embracing new technologies, those efforts are known, amongst others, as Industrie 4.0, Cyber Physical Production Systems (CPPS), or Industrial Internet of Things (IIoT). [1]

A major requirement for leveraging the full potential of Industrie 4.0 applications is the utilization of big data and data analytics methods in production systems. These methods are used to reveal otherwise unknown knowledge, enable process improvements, and increase the overall equipment efficiency (OEE). In modern automated production systems, the generated data shares many similarities with big data, defined via the four V's (volume, variety, velocity, and value) [2]. Several factors handicap automated data analysis in the field of automation. Especially the multitude and heterogeneity of data sources, formats, and protocols due to long lifecycles (up to 30 years) in the production environment pose a challenge (variety). In addition, large amounts of historic data (volume) have to be combined with constantly streamed data from the plant in order to make decisions (value) based on the analysis results in time (velocity). Classical data analysis approaches are not applicable in this heterogeneous automation context. Therefore, new, innovative system architectures for applying big data techniques in automated production systems have to be developed. [3–5]

These difficulties become evident using an example from process industry: A multitude of sensors continuously collect process data, which is stored in databases mainly for documentation purposes. A manufacturing execution system (MES) is used to manage data concerning resource planning and order execution. Moreover, a shift book contains information about operators responsible for surveying the mode of operation and incidents happening during their shifts. Adding additional complexity, quality and maintenance data may be stored in other systems or databases. Together, they form a complex network of interwoven IT systems, based in different physical locations, relying on different, often incompatible data formats. Extracting knowledge from this heterogeneous setup is difficult and can often impossible without a huge manual effort carried out by experts. Consequently, an architecture for unifying data access could greatly enhance the possibilities and impact of data analysis in production environments. This could be achieved by including all relevant sources and providing their data for analysis tools and enabling ubiquitous computing. In this context, we consider the term "architecture" to be defining the description of the overall system layout based on principles and rules in order to describe its construction, enhancement and usage. This definition is compliant with the Reference Architecture Model Industrie 4.0 (RAMI 4.0) [6].

We therefore suggest that the implementation of such a software architecture in production systems is desirable and possible, simplifying data acquisition, aggregation, integration and warehousing, as well as providing data for analysis. This paper describes a generic architecture that can be applied to various scenarios and shows its concrete use and practicability in a lab-scale production system. It pays special attention to the multitude of requirements arising from automated production systems, legacy systems, heterogeneous sources, and data processing,

This contribution is an extended and adapted version of the contribution presented at the 2017 IEEE International Conference on Industrial Technology (ICIT 2017) [7]. In addition to the original version, the literature review is expanded and a prototypical implementation is added.

The remainder of the paper is structured as follows: We first derive requirements for a system architecture to support Industrie 4.0 principles, then evaluate how other authors fulfill them, and identify a research gap. After deriving a concept for a new architecture, we evaluate it, using expert interviews and a prototypical lab-scale implementation. The last part summarizes the findings and provides an outlook for further applications and fields of research.

2 Requirements for a System Architecture to Support Industrie 4.0 Principles

One of the main goals of Industrie 4.0 is the optimization of the manufacturing process based on algorithms and data to increase the OEE [3]. This can be achieved by gaining a better understanding of complex procedures inside the plant and thus reducing maintenance and downtimes. Therefore, new ways of knowledge discovery have to be provided and the vertical integration of the production process has to be enhanced. In order to establish a new system architecture to support Industrie 4.0 principles, several requirements have to be considered.

Supporting various data sources, including legacy systems, is one of the key aspects for successfully implementing a common architecture (requirement R1). Characteristic for the data landscape of a manufacturing enterprise is its heterogeneity and variety of systems. Currently, each layer of an enterprise typically operates using a multitude of layer-specific protocols and formats. These communication channels were often not explicitly designed to operate with other tools. Long lifespans in automation industry ensure the presence of legacy devices that prohibit disruptive changes. [8–10] The new architecture thus has to be able to operate without interfering with the mode of operation of existing systems. Moreover, integration of tools and machines has to be possible regardless of their technology or capabilities [11–13]. A support for various sources can be achieved by defining interfaces, which have to be implemented by data adapters, transferring specific protocols and data formats into a common one.

Thus, we derive as requirement R2 a common data model. As Hashem et al. [2] stated, data staging is an open research issue that needs more attention in the future. Introducing a reference data model [5, 14] can greatly reduce manual work for data integration if it serves towards a common understanding of the data from all involved systems. An exemplary data model is the ontology defined in ISO 15926 Part 8 [15], standardizing data exchange between different IT systems, or the Common Information Model (CIM) defined in the standards EN 61968 and EN 61970 [16, 17] for electrical distribution systems.

Having successfully integrated various sources and transferred data to a common model, dealing with data of different timeliness and message sizes has to be considered. A suitable architecture should be able to process both historic and near real-time data cor-

rectly (R3). One the one hand, some sources may constantly send small messages containing information about single variables from within the production process. On the other hand, other sources only send a single message with enormous message size (batch message) when queried, containing for example order information. Some messages may contain near real-time information about current processes, others time-insensitive contents. This implies that the architecture has to be able to extract knowledge from constantly arriving message streams and huge amounts of batch messages. [18] Marz and Warren [19] suggest the so-called lambda architecture. The lambda architecture consists of different layers, namely a speed layer for real-time tasks, a batch layer for accessing historical data from a storage, and a serving layer, combining the results of batch and speed layer. Being capable of handling hybrid processing at near- real-time, their concept is one example for achieving this task.

In order to be applicable for a wide range of use cases, the architecture has to be able to handle various analysis methods and thus provide interfaces for including different GUIs and HMIs, export possibilities, and query tools (R4). In order to simplify data retrieval, the architecture should be able to provide users the collected data at the right level of abstraction (depending on the request) [4]. Also, due to the nature of Big Data, a single form of data visualization is not sufficient. It is problematic to find user-friendly visualizations. Thus, new techniques and frameworks should be includable into the architecture, requiring well-defined interfaces for queries. [20] Begoli and Horey [21] suggest the usage of open, popular standards, exposition of results via web interfaces, and the adoption of lightweight, web-oriented architectures. They also confirm that many different visualizations across many datasets are necessary to extract knowledge from the data.

Not only in-company analysis can be deducted, but to leverage the full potential of the data, a cross-organizational data transfer and analysis process is necessary. This is why as requirement R5 a support for anonymized data transfer across organizational borders is deduced. Manufacturers could develop models for lifetime prediction, improve production lines, and increase the OEE by gathering data from their shipped product. However, once a machine is shipped to the customer, the manufacturers of production machines rarely can access relevant sensor data to predict lifespan and possible failures. Revealing causes for hardware failures can be supported by collecting datasets across corporate boundaries, and analyzing process, device, and servicing data. Relying on more data sources and more data contents, lifespan predictions can become more accurate. Measures for improving the OEE can be deducted through the utilization of better models. [22] In order to guarantee privacy protection and security, those datasets have to be anonymized and transferred via a secure connection to allow analysis by selected personnel. Developing efficient and secure ways for big data exchange is still an open research issue. [2, 23] Protection of trade secrets is a very important aspect for every company, making data transfer across organizational borders a highly sensitive topic. [2]

All derived requirements are summarized in **Table 1**.

Table 1. Table of Requirements for Industrie 4.0 System Architectures.

ID	Description
R1	**Support of various data sources, including legacy systems**. Handling of heterogeneous data from all relevant sources, being able to include existing legacy systems.
R2	**Usage of a common data model**. A common understanding of the aggregated data is necessary to simplify the analysis task. This implies the definition of a common data model and can greatly simplify the communication between different data sources, services and business units.
R3	**Processing of historic and near real-time data.** Being able to handle both batch messages and stream data is an integral requirement for processing messages with different timeliness.
R4	**Support different analysis methods and tools.** Different levels of abstraction and support for various analysis tools are important to support users in detecting patterns among the data.
R5	**Support anonymized data transfer across organizational borders** By including data from various parties into analysis, life span predictions can become more accurate and models for determining possible failures can become more sophisticated. In order to guarantee privacy protection and protection of trade secrets, data transfer has to be done in an anonymized and secure way.

3 State-of-the-Art of Industrie 4.0 System Architectures

Several reference architectures exist in the context of Industrie 4.0 and the Industrial Internet of Things (IIoT). The most important ones are the German Reference Architecture Model Industrie 4.0 (RAMI 4.0) [6], the American Industrial Internet Reference Architecture (IIRA) [24] and the draft international standard ISO/IEC CD 30141 [25] for the Internet of Things Reference Architecture (IoT RA). These reference architectures provide an abstract, technology-neutral representation of an IIoT system and rules for the development of an actual architecture. Therefore, they feature an abstract description, which has to be adopted in order to represent the specific characteristic of an actual system.

The concept of the Enterprise Service Bus (ESB) was proposed by Chappell [26]. The ESB relies on a communication and integration backbone to connect different applications and technologies in an enterprise. It employs web service technologies and supports various communication protocols and services. One of the main goals of the ESB is to include various heterogeneous sources and services (R1). This is achieved by using a common data model (R2) for passing messages over the central bus. The design of the ESB also supports different analysis methods (R4), since existing APIs can be used. Anonymized data transfer across organizational borders (R5) is not covered by the ESB. Different data processing ways (R3) are not explicitly mentioned in the description of the ESB and would

require special attention during the design and implementation of an Enterprise Service Bus.

The Automation Service Bus (ASB), introduced by Moser, Modrinyi and Winkler [27] is based on the concept of the ESB, but narrows down the scope to engineering tasks in the field of mechatronics development. The ASB uses a common data model (R2) to provide a platform for all disciplines taking part in the development of mechatronic systems. Since the ASB focuses on engineering phase of the asset lifecycle, operational data is not supported (R1). A special consideration of legacy systems is not mentioned, neither whether data adapters exist. Different processing ways (R3) and anonymized data transfer between enterprises (R5) do not exist in the concept. Since the ASB is tailored towards the developed Engineering-Cockpit as a user interface, other analysis methods (R4) are not supported.

The Namur Open Architecture (NOA) presented by Klettner et al. [28] is an additive structure to the conventional production pyramid [29]. Its structure allows an open information exchange over a secondary communication channel between not-neighboring automation layers and a secure backflow from an IT environment into process control. The Namur Open Architecture specifies how information is transferred from the core process control to plant specific monitoring and optimization applications. This is achieved by using open and vendor-independent interfaces. A special interest of the NOA is to support various existing systems and data sources (R1). The architecture can be connected to various applications and means of analysis (R4). A cross-organizational data transfer is intended via the “Central Monitoring + Optimization” part, however, anonymization is not mentioned (R5). NOA describes two channels for data transfer from shop floor devices to the analysis part (“Central M+O”). The direct path could be used to transfer soft real-time data, whereas open interfaces could be used to process batch data (R3). The NOA relies on a common data model (R2) serving as unified space of information for process data.

The PERFoRM project [30] focuses on seamless production system reconfiguration, combining the human role as flexibility driver and plug-and-produce capabilities. It is compliant with legacy systems by using proper adapters to translate all data into a system-wide language. Special focus is put on integrating systems from all stages of the production pyramid using suitable wrappers, interfaces and adapters (R1). Adapters transfer data models of legacy devices into a common data model (R2). The architecture’s middleware ensures reliable, secure and transparent interconnection of the various hardware devices and software application. It therefore follows a service-oriented principle, i.e. offering functionalities as services, which can be discovered and requested by other components. Since it was designed in a human centered way, various HMIs and analysis methods are supported (R4). The proposed middleware is independent from a certain network technology; various devices implementing the defined interface can be used for analyzing results. A way for transferring data between companies and anonymizing them (R5) is not described by the PERFoRM architecture. The different processing ways for batch data and soft real-time data (R3) are not taken into consideration by the concept.

The Line Information System Architecture (LISA), proposed by Theorin et al. [11], is an event-driven architecture featuring loose coupling and a prototype-oriented information model. LISA puts special focus on integration of devices and services on all levels (R1), reducing effort of hardware changes and delivering valuable aggregated information in the

form of KPIs. It uses simple messages from various kinds of devices that are translated via data adapters to fit the LISA message format (R2) and then sent to an Enterprise Service Bus (ESB). The ESB offers publish-subscribe functionality and forwards the messages to all interested receivers. Knowledge extracted from the messages can be used for online monitoring, control, optimization, and reconfiguration. By using data adapters, various analysis methods can be integrated (R4). As not being based on point-to-point communication, the event-driven architecture can be easily reconfigured. Since new event creators only need to know that the event occurred, but not who is receiving the message or how it has to be processed, loose-coupling is achieved. Real-time data will be available directly from the events and historic data can be queried from a suitable storage system (R3). Data exchange across organizational borders is not intended by the architecture (R5).

Hufnagel and Vogel-Heuser [31] present a concept for facilitating the capturing of distributed data sources, relying on the ESB-principle. Their model-based architecture uses data mapping and adapters to transfer data of various sources (R1) into a common data model (R2). It is able to handle batch data and real-time data from different systems (R3). However, the concept does not focus on analysis (R4), or sharing and anonymizing data (R5).

The presented concepts and the degree of requirement fulfillment are summarized in **Table 2.** None of the existing approaches fulfills all derived requirements.

Table 2. Classification Matrix for Existing Industrie 4.0 System Architecture Concepts.

Concept	R1	R2	R3	R4	R5	Implemented
ESB [26]	+	+	O	+	O	+
ASB [27]	-	+	-	-	-	+
NOA [28]	+	+	O	+	O	+
PERFoRM [30]	+	+	-	+	-	-
LISA [11]	+	+	+	+	-	+
Hufnagel and Vogel-Heuser [31]	+	+	+	-	-	-

Legend: + fulfilled, O *partly fulfilled, - not fulfilled*

4 Concept of a Unified Data Transfer Architecture (UDaTA) in Automated Production Systems

In the following, the concept of the Unified Data Transfer Architecture (UDaTA) will be derived from the aforementioned requirements, paying special attention to its suitability for different use cases. Focus is put on defining the overall concept technology-independently, meaning the specific technologies for an implementation can be adjusted to

fulfill the requirements of a given use-cases (e.g. usage of MQTT instead of OPC UA or Kafka [32] instead of an enterprise service bus).

In order to support different analysis methods, various tools (R4) and legacy devices (R1), standardized interfaces are necessary. Relying on a layered structure with well-defined interfaces simplifies reconfiguration and adoption to a variety of use cases. The architecture differentiates between layers for providing raw data, analyzing data, and displaying data. Connecting data sources with its destinations, the so-called *Data Management and Integration Broker* transfers and routes data between the components and layers of UDaTA. This setting allows for both the connection of existing legacy applications as well as newly added tools through interfaces.

Requirement 3 demands for the ability to process both historical and near real-time data. Therefore, UDaTA features a central data storage for saving data and providing it for later analysis. Real time data from data sources is streamed by the broker to the data storage and made available there. Depending on the use-case (e.g. number of sources and message intensity), the data storage can be a relational or non-relational database. Components of the analysis or displaying layer can request data from the storage via the broker, using the aforementioned standard interfaces. Handling all types of data, the central data storage ensures a wide availability of data for all layers. Making not only historic data, but also real-time data available, the Data Management and Integration Broker can stream live data to subscribers of the upper layers, as described by the lambda architecture [19]. This can be important for live-monitoring and optimization operations executed by the analysis layer.

For enabling data transfer across organizational borders (R5), an access control and anonymization component is necessary. Especially, when working with data from other organizational units or companies, the privacy and integrity of data become very important issues. Leaking data, which was not supposed to be transferred, must be avoided. Hence, the broker features an access control and anonymization layer, guaranteeing only approved data access. Before transferring content, data may be anonymized or access restricted, depending on the requesting block and its security clearance.

A common data model (R2) is essential in order to describe the represented system and its data in a unified way and in a language that is understood by all layers and components. The data model of UDaTA has to include representations of the raw data, additional metadata (enriching raw data with information about units of measurement, associated devices, and so on), previously learned and preconfigured models, operator knowledge, parameter sets, and configurations of the service blocks. Each block can work with a subset of the overall data model for carrying out its operations. Data that does not match the common model has to be transferred by adapters in order to be compliant with the other data as presented in [33, 34]. Wrappers can be used to encapsulate third-party applications and provide standardized, compatible interfaces. Those mappings and adaptions have to be carried out by hand, when dealing with a new data model or changes during the asset lifecycle. Especially for legacy blocks, the effort for translating data can sometimes be high, but the benefits of dealing with only one common data format, like easy plug-in of new blocks, as well as high compatibility and increased flexibility, are significant. This is also a reason why a parallel rollout of UDaTA to existing systems and a stepwise adaption is suggested as deployment strategy.

Fig. 1. gives a representation of the Unified Data Transfer Architecture, reflecting one company or organizational structure. Each company or structure can have its own slice of the architecture deployed, allowing a communication over the broker and the analysis of data stored in a different location. Several instances of the broker can communicate with each other and exchange data.

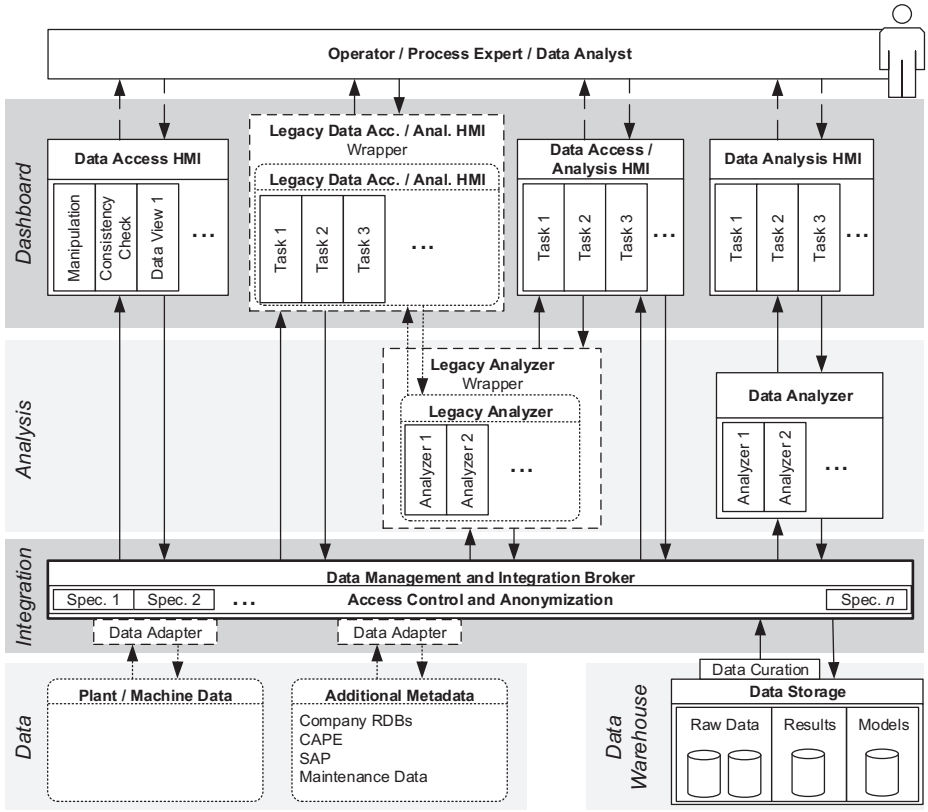

Fig. 1. Schematic Structure of the Unified Data Transfer Architecture (UDaTA).

5 Evaluation

The proposed architecture was evaluated in two ways. At first, an expert evaluation was carried out. Therefore, technical experts were interviewed about an adapted version of the architecture for their specific use-cases. The use-cases originate from different fields, namely process industry and discrete manufacturing. In addition, questionnaires were used to query specific aspects of the architecture.

Furthermore, a prototypical, lab-scale demonstration was implemented using concrete technologies in order to reflect the realization for a specific use-case.

5.1 Expert Evaluation

Semi-structured interviews with technical experts were carried out for two distinct use-cases from the field of process industry and discrete manufacturing. Both use-cases are characterized by a multitude of existing legacy systems, complex system layouts and interactions between software components, as well as a high degree of heterogeneity in the used communication channels (e.g. different field buses) and data formats. For the interviews, the generic UDaTA was adapted for the specific use-cases, reflecting the specific requirements of the field of application. It was evaluated whether there is a need for a unified architecture for data access under the given boundary conditions. In addition, the interviewees were asked about the shortcomings of their current system architectures and if the proposed UDaTA has the potential to solve these disadvantages and inefficiencies. The interviews were supported by questionnaires for capturing specific needs and requirements in the field of application.

In both existing solutions, data is often aggregated and integrated manually by a process expert. Therefore, data from various sources is combined, e.g. written maintenance logs are digitized and combined with the historic process data queried from a plant information management system (PIMS). This task is highly time-consuming and error-prone, especially for repetitive analysis.

Existing legacy systems with proprietary interfaces or the possibility to export data only file-based, e.g. export to csv-files, further complicate the existing solutions and lead to countless point-to-point connections and tailor-made solutions for specific applications.

For both use-cases, the interviewed experts expressed the need for a unifying and innovative architecture. Therefore, the characteristics and flexibility of UDaTA were evaluated positively in both cases. Furthermore, the experts highlighted the importance of access control and anonymization, as well as the significance of cross-organizational data exchange for information extraction from local raw data.

As problematic aspects of UDaTA the definition of an accepted common information model as well as the high effort for a first implementation were mentioned. Overcoming the issue of the definition of a data model is challenging. It can be carried out use-case specific by an interdisciplinary team of experts focusing on the specific needs for the use-case. Alternatively, it could be promoted by a standardization body, for instance following the example of the common information model (CIM) in the energy distribution sector [16, 17]. For the field of automated and cyber-physical production systems, there is an immediate need for such an accepted meta-model. The CIM reflects the evolution of an accepted meta-model unifying the data understanding between different parties and should act as a guidance for such a model in the field of automation.

In order to overcome the complex introduction phase of UDaTA, a stepwise migration is proposed, leaving the existing ISA 95 layout intact. This allows to deploy the architecture in parallel to the existing infrastructure and to integrate the different processes of the production plant over time, therefore benefiting from already migrated applications while leaving the vital, production-relevant infrastructure untouched.

5.2 Prototypical Lab-Scale Implementation

In order to verify the practicability of the concept, a prototypical implementation is deployed on a lab-scale. To simulate a heterogeneous production environment, a bench-scale platform, called extended pick and place unit (xPPU) [35], is used as the main data source for production data. The xPPU is able to sort, pick, and place work pieces, using amongst others cranes, conveyor belts, and a multitude of sensors. As secondary data source, exemplary order data from a csv file is read. The data is sent to a message broker, translated into a common data format and stored in a relational database. Analyzers can send requests for batch data to the broker. These requests are forwarded to the database, the results are anonymized if necessary, and provided to analyzers. In addition, analyzers can subscribe to live data originating from the sources. In order to replicate the heterogeneity of the production environment, various different operating systems, as well as programming languages to realize the separate applications, are used. The whole setup is depicted in **Fig. 2**, showing hardware components, operating systems, software, and exemplary data flows among the elements of the architecture. The setup is described in detail in the following paragraphs.

In the prototypical setup, the bus couplers of the xPPU are communicating via EtherCAT with a Beckhoff CX9020 PLC that runs on Windows 7 CE and TwinCAT 3. This PLC makes its process data available over two different communication channels, namely MQTT messages and an OPC-UA server.

MQTT [36] is a publish-subscribe-based, lightweight messaging protocol that is suitable for Machine-to-Machine (M2M). Using the TwinCAT function block “Tc3_IotBase”, MQTT-client functionality becomes available on the PLC. This way, messages with topics like “EnergyMonitoringHardware/CurrentPressure/Int” or “LightGrid/EmergencyStop/Bool” are sent every cycle (currently 10ms). These messages contain the values of the respective variables of the xPPU.

Furthermore, the PLC runs an OPC UA server that provides the states of other selected variables to clients. In our case, a Raspberry Pi 3 is used as a client for the PLC. It operates with Raspbian running an OPC UA client using the .Net Standard reference implementation by the OPC Foundation [37]. Furthermore, it subscribes to a number of variables and translates the received data into the common data model. This data is then sent to the connected broker.

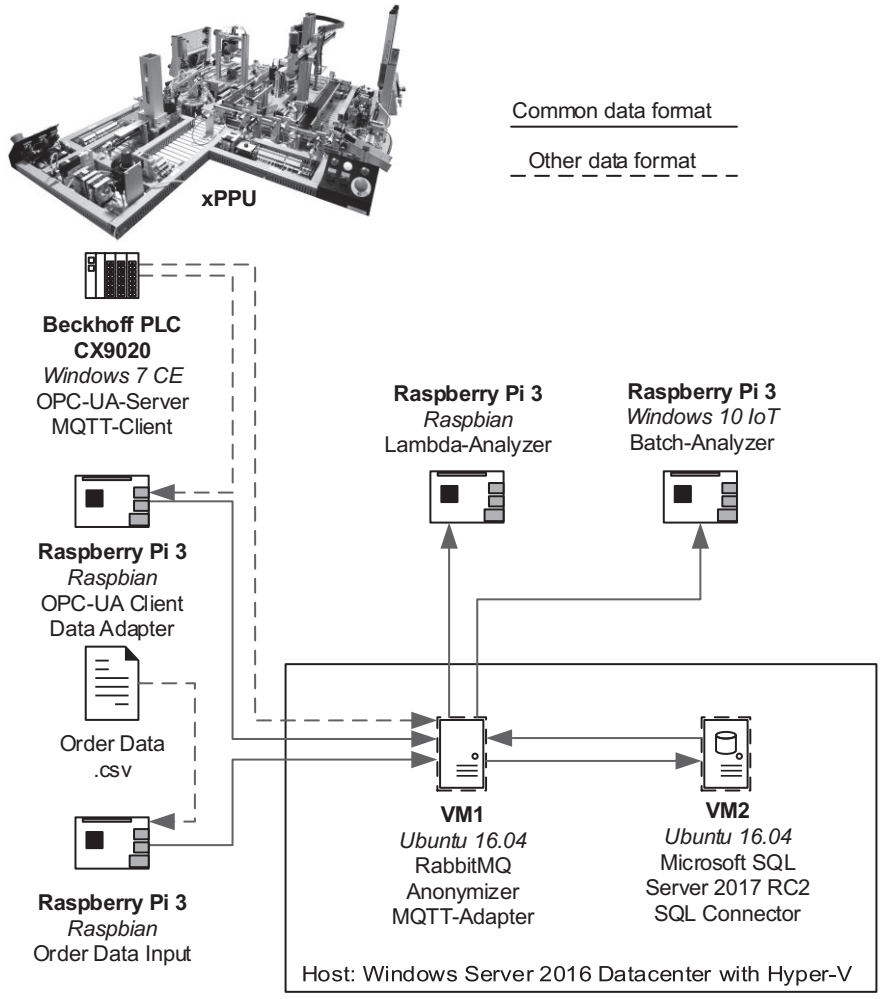

Fig. 2. Hardware and Software Setup of the Prototypical Implementation for the xPPU.

To simulate the processing of batch data, a csv-file containing hypothetic order information is read by a Java program on another Raspberry Pi 3 with Raspbian, translated directly into the common data format and sent to the broker.

Using Hyper-V on a Windows Server 2016 Datacenter host, that is equipped with a Core i7-6700 CPU and 16 GB RAM, we installed Linux Ubuntu 16.04 LTS x64 on two virtual machines (VM).

On the first VM, we installed the open source broker RabbitMQ [38] in version 3.6.11, as well as a .Net Core 2.0 based Anonymizer and Translator for MQTT messages. RabbitMQ works with exchanges, to which messages can be sent. An exchange can be connected to several queues that store messages until they are polled or being subscribed to.

This setup allows for a distribution of messages to the correct destinations. The Translator receives MQTT messages sent by the PLC and translates those into the common data format. If the requesting analyzer has only limited access rights, the Anonymizer for instance changes data and time values to a relative scale in order to minimize information leakage.

The second VM runs the database components. In is this case they compromise a Microsoft SQL Server 2017 RC2 and a .NET Core 2.0 based SQL Connector that receives messages and queries from the broker and handles database communication using Microsoft Entity Framework Core 2.0.

Using two more Raspberry Pis 3, one with Raspbian, one with Windows 10 IoT, instances of .NET Core 2.0 based analyzers are executed. These can subscribe to live data from sources, request data from the database, perform calculations on the data, send calculated data to the broker, or listen to the results of other analyzers.

With this setup, the feasibility of the implementation of an architecture for unifying data transfer in automated production systems is demonstrated. Relying on platform independent technologies like .NET Core 2.0, Java and an open source message broker, that can be executed on Windows, Linux, and macOS, the implementation can be rolled-out in heterogeneous IT environments (R1). By using adapters and translators, the transformation of messages into a common data model is carried out (R2). It is possible to connect arbitrary analyzers (R4) (including legacy components) to the open source message broker RabbitMQ, since it provides clients and developer tools for many programming languages (such as Java, .Net, Ruby, Python, PHP, JavaScript, Objective-C, C, C++), which was also demonstrated in the heterogeneous environment chosen for the demonstration. Analyzers are able to access both historic and live data (R3). Using different roles with different access rights on the broker, data security is ensured. Moreover, as data can be automatically anonymized if necessary, also sensitive information can be shared. Connecting brokers at different physical locations is possible with the Shovel plugin [39] of RabbitMQ (R5). For the realization of UDaTA, it is emphasized, that the selected technologies, languages, or brokers are only of subordinate relevance; the shown prototypical implementation is only one possible solution for this specific use-case.

In order to validate the feasibility of the broker for a large number of incoming and outgoing messages, a time measurement of the delivery times from start to destination is performed. Before carrying out the tests, the internal clocks of each device were synchronized. Afterwards, timestamps are added to the message headers and the timespan it took to deliver the message is calculated. Performing these tests with low message intensity (~2 msg/ sec published), middle intensity (~80 msg/ sec published) and high intensity (~1650 msg/ sec published) for several minutes, no influence of the message intensity on the delivery times is observed. The average message times varied for all intensities between 2.5 and 10.0 milliseconds. These absolute times, however, have to be interpreted with caution since it cannot be guaranteed that the clocks were perfectly synced in the range of milliseconds. What can however be stated is that no evidence of prolonged message times due to higher message intensity is measured for the given setup. Therefore, RabbitMQ is a suitable message broker for providing live data to analyzers in a Unified Data Transfer Architecture with near real-time requirements under the given load scenario.

6 Conclusion and Outlook

Applying big data techniques in the field of manufacturing is currently strongly handicapped due the multitude of protocols and data formats used by legacy devices deployed to the field. Manual data acquisition from closed, proprietary system and subsequent integration of data by experts are often the only chance to access the vast amount of measured data. However, there is a need for making this data automatically available in order to gain information from it, especially with the rise of the ideas of the industrial internet of things (IIoT) and Industrie 4.0. For Cyber-physical systems (CPS) and cyber-physical production systems (CPPS), the transparency of information, as well as big data analytics play a major role. New, flexible architectures are required in order to make this information accessible and apply big data in the field of automation.

This contribution presents a conceptual architecture named UDaTA for data acquisition, integration, and handling from the field device layer up to business applications. Therefore, it provides mechanisms for vertical, as well as horizontal integration. A strong effort is put on the unification of data access and transport for abstracting the complexity of the involved systems. Reference models for IIoT, like the German RAMI4.0 or American IIRA, lay the cornerstones for such an architecture, but due to their generic nature do not capture application specific aspects. Existing concepts for data acquisition and integration in the field of automation often lack the consideration of cross-company data exchange for collaboration and openness of the interfaces. The conceptualized architecture uses a middleware-approach to make the data available and minimizes the number of data transformations between the involved systems by proposing the use of a common information model (CIM).

The concept was evaluated with expert interviews showing an apparent need for the implementation of such an architecture. Moreover, using a bench-scale production system, it was possible to show that UDaTA can be prototypically implemented and is able to transfer and handle data from heterogeneous sources. Acting as middleware component, the open source message broker RabbitMQ received data input from MQTT and OPC UA sources, which was transferred via adapters into a common information model. Both, access to streamed live data and historic batch data, could be demonstrated and opened new possibilities for data analysis.

In order to prove its suitability for real production environments, further testing with larger prototypes and more data sources is required. Special focus has to be put on the formulation of a CIM that is suitable for generic use cases and allows translation from various data formats. Other technologies for the middleware, for example Apache Kafka, OPC UA or DDS-Systems, should also be evaluated and their eligibility for real world use cases compared. Even more important, real analysis of the data and usage of the newly gained knowledge is required. Next steps should focus on using the acquired data for the analysis process.

Acknowledgment

This work is part of the two projects IMPROVE and SIDAP.

IMPROVE (improve-vfof.eu) has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 678867.

In addition, SIDAP (www.sidap.de) has received funding by the German Federal Ministry for Economic Affairs and Energy (BMWi) under the grant number 01MD15009F.

References

1. Bauer, H., Baur, C., Camplone, G.: Industry 4.0. How to Navigate Digitization of the Manufacturing Sector. tech. rep., McKinsey Digital (2015)
2. Hashem, I.A.T., Yaqoob, I., Anuar, N.B., Mokhtar, S., Gani, A., Ullah Khan, S.: The Rise of "Big Data" on Cloud Computing. Review and Open Research Issues. Information Systems 47, 98–115 (2015)
3. Vogel-Heuser, B., Hess, D.: Guest Editorial Industry 4.0–Prerequisites and Visions. IEEE Trans. Automat. Sci. Eng. 13, 411–413 (2016)
4. Cecchinel, C., Jimenez, M., Mosser, S., Riveill, M.: An Architecture to Support the Collection of Big Data in the Internet of Things. In: Zhang, L.-J. (ed.) IEEE World Congress on Services (SERVICES), 2014, pp. 442–449 (2014)
5. Jirkovsky, V., Obitko, M., Marik, V.: Understanding Data Heterogeneity in the Context of Cyber-physical Systems Integration. IEEE Trans. Ind. Inf. 13, 660–667 (2017)
6. Deutsches Institut für Normung e.V. (DIN): Reference Architecture Model Industrie 4.0 (RAMI4.0) (2016)
7. Trunzer, E., Kirchen, I., Folmer, J., Koltun, G., Vogel-Heuser, B.: A Flexible Architecture for Data Mining From Heterogeneous Data Sources in Automated Production Systems. In: 2017 IEEE International Conference on Industrial Technology (ICIT), pp. 1106–1111 (2017)
8. Delsing, J., Eliasson, J., Kyusakov, R., Colombo, A.W., Jammes, F., Nessaether, J., Karnouskos, S., Diedrich, C.: A Migration Approach Towards a SOA-based Next Generation Process Control and Monitoring. In: IECON 2011 - 37th Annual Conference of the IEEE Industrial Electronics Society, pp. 4472–4477 (2011)
9. Vogel-Heuser, B., Kegel, G., Bender, K., Wucherer, K.: Global Information Architecture for Industrial Automation. Automatisierungstechnische Praxis (atp) 51, 108–115 (2009)
10. The Industrial Internet of Things. Volume G5: Connectivity Framework (2017)
11. Theorin, A., Bengtsson, K., Provost, J., Lieder, M., Johnsson, C., Lundholm, T., Lennartson, B.: An Event-driven Manufacturing Information System Architecture for Industry 4.0. International Journal of Production Research 55, 1297–1311 (2016)
12. International Organization for Standardization (ISO): Information technology – Internet of Things Reference Architecture (IoT RA) (2016)
13. Delsing, J., Eliasson, J., Kyusakov, R., Colombo, A.W., Jammes, F., Nessaether, J., Karnouskos, S., Diedrich, C.: A Migration Approach Towards a SOA-based Next Generation Process

Control and Monitoring. In: IECON 2011 - 37th Annual Conference of the IEEE Industrial Electronics Society, pp. 4472–4477 (2011)
14. Castano, S., Antonellis, V. de: Global Viewing of Heterogeneous Data Sources. IEEE Trans. Knowl. Data Eng. 13, 277–297 (2001)
15. Industrial Automation Systems and Integration – Integration of Life-cycle Data for Process Plants Including Oil and Gas Production Facilities – Part 8: Implementation Methods for the Integration of Distributed Systems: Web Ontology Language (OWL) Implementation ISO 15926-8
16. European Committee for Electrotechnical Standardization (CENELEC): Application Integration at Electric Utilities – System Interfaces for Distribution Management – Part 1: Interface Architecture and General Recommendations (IEC 61968-1:2012) (2013)
17. European Committee for Electrotechnical Standardization (CENELEC): Energy Management System Application Program Interface (EMS-API) - Part 1: Guidelines and General Requirements (IEC 61970-1:2005) (2007)
18. Casado, R., Younas, M.: Emerging Trends and Technologies in Big Data Processing. Concurrency Computat.: Pract. Exper. 27, 2078–2091 (2015)
19. Marz, N., Warren, J.: Big Data. Principles and Best Practices of Scalable Real-time Data Systems. Manning, Shelter Island (2015)
20. Fan, W., Bifet, A.: Mining Big Data: Current Status, And Forecast to the Future. SIGKDD Explor. Newsl. 14, 1–5 (2012)
21. Begoli, E., Horey, J.: Design Principles for Effective Knowledge Discovery from Big Data. In: Babar, M.A. (ed.) Joint Working IEEE/IFIP Conference on Software Architecture (WICSA) and European Conference on Software Architecture (ECSA), 2012, pp. 215–218 (2012)
22. Institute of Automation and Information Systems, Technical University of Munich: SIDAP. Skalierbares Integrationskonzept zur Datenaggregation, -analyse, -aufbereitung von großen Datenmengen in der Prozessindustrie, http://www.sidap.de/
23. Lu, X., Li, Q., Qu, Z., Hui, P.: Privacy Information Security Classification Study in Internet of Things. In: 2014 International Conference on Identification, Information and Knowledge in the Internet of Things, IIKI 2014, pp. 162–165. IEEE (2014)
24. The Industrial Internet of Things. Volume G1: Reference Architecture (2017)
25. International Organization for Standardization (ISO): Information technology – Internet of Things Reference Architecture (IoT RA) (2016)
26. Chappell, D.: Enterprise Service Bus. O'Reilly Media, Inc. (2004)
27. Moser, T., Mordinyi, R., Winkler, D.: Extending Mechatronic Objects for Automation Systems Engineering in Heterogeneous Engineering Environments. In: Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies & Factory Automation (ETFA 2012), pp. 1–8 (2012)
28. Klettner, C., Tauchnitz, T., Epple, U., Nothdurft, L., Diedrich, C., Schröder, T., Goßmann, D., Banerjee, S., Krauß, M., Latrou, C., et al.: Namur Open Architecture. Die Namur-Pyramide wird geöffnet für Industrie 4.0. Automatisierungstechnische Praxis (atp) 59, 20–37 (2017)
29. European Committee for Electrotechnical Standardization (CENELEC): Enterprise-control System Integration – Part 1: Models and Terminology (IEC 62264-1:2013) (2014)
30. Leitao, P., Barbosa, J., Pereira, A., Barata, J., Colombo, A.W.: Specification of the PERFoRM Architecture for the Seamless Production System Reconfiguration. In: Proceedings of the IECON2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society, pp. 5729–5734 (2016)
31. Hufnagel, J., Vogel-Heuser, B.: Data Integration in Manufacturing Industry: Model-based Integration of Data Distributed From ERP to PLC. In: 2015 IEEE 13th International Conference on Industrial Informatics (INDIN), pp. 275–281 (2015)

32. Wang, Z., Dai, W., Wang, F., Deng, H., Wei, S., Zhang, X., Liang, B.: Kafka and Its Using in High-throughput and Reliable Message Distribution. In: 2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS), pp. 117–120 (2015)
33. Karnouskos, S., Bangemann, T., Diedrich, C.: Integration of Legacy Devices in the Future SOA-based Factory. IFAC Proceedings Volumes 42, 2113–2118 (2009)
34. Derhamy, H., Eliasson, J., Delsing, J.: IoT Interoperability - On-demand and Low Latency Transparent Multi-protocol Translator. IEEE Internet Things J., 1 (2017)
35. Vogel-Heuser, B., Legat, C., Folmer, J., Feldmann, S.: Researching Evolution in Industrial Plant Automation. Scenarios and Documentation of the Pick and Place Unit (2014)
36. International Organization for Standardization (ISO): Information Technology - Message Queuing Telemetry Transport (MQTT) v3.1.1 (2016)
37. OPC Foundation, https://github.com/OPCFoundation/UA-.NETStandardLibrary
38. Pivotal Software Inc.: RabbitMQ, https://www.rabbitmq.com/
39. Pivotal Software Inc.: Shovel plugin, https://www.rabbitmq.com/shovel.html