



System architectures for Industrie 4.0 applications

Derivation of a generic architecture proposal

Emanuel Trunzer¹  · Ambra Calà^{2,3} · Paulo Leitão⁴  · Michael Gepp² · Jakob Kinghorst¹ · Arndt Lüder³ · Hubertus Schauerte⁵ · Markus Reifferscheid⁵ · Birgit Vogel-Heuser¹ 

Received: 20 December 2018 / Accepted: 14 April 2019
© The Author(s) 2019

Abstract

Industrie 4.0 principles demand increasing flexibility and modularity for automated production systems. Current system architectures provide an isolated view of specific applications and use cases, but lack a global, more generic approach. Based on the specific architectures of two EU projects and one German Industrie 4.0 project, a generic system architecture is proposed. This system architecture features the strengths of the three isolated proposals, such as cross-enterprise data sharing, service orchestration, and real-time capabilities, and can be applied to a wide field of applications. Future research should be directed towards considering the applicability of the architecture to other equal applications.

Keywords Cyber-physical systems · System architecture · Middleware · Interoperability · Reconfigurability · Industrie 4.0

PERFoRM has received funding from the European Unions Horizon 2020 research and innovation programme under Grant agreement no. 680435. IMPROVE has received funding from the European Unions Horizon 2020 research and innovation programme under Grant agreement no. 678867. BaSys 4.0 has received funding from the German Federal Ministry of Education and Research (BMBF) under Grant no. 01IS16022N.

✉ Emanuel Trunzer
emanuel.trunzer@tum.de

Ambra Calà
ambra.cala.ext@siemens.com

Paulo Leitão
pleitao@ipb.pt

Michael Gepp
michael.gepp@siemens.com

Jakob Kinghorst
j.kinghorst@tum.de

Arndt Lüder
arndt.lueder@ovgu.de

Hubertus Schauerte
hubertus.schauerte@sms-group.com

Markus Reifferscheid
markus.reifferscheid@sms-group.com

1 System architectures in Industrie 4.0

Automated production systems (aPS) combine machines and production units from different manufacturers, as well as tools for data warehousing and monitoring. Due to the high amount of digitalization and the integration of cooperating, self-aware subsystems, aPS can be seen as cyber-physical production systems (CPPS) [1]. They form connected units by offering their services through service-oriented

Birgit Vogel-Heuser
vogel-heuser@tum.de

- 1 Institute of Automation and Information System, Technical University of Munich, Boltzmannstr. 15, 85748 Garching, Germany
- 2 Siemens AG Corporate Technology, Günther-Scharowsky-Str. 1, 91058 Erlangen, Germany
- 3 Center for Distributed Systems, Otto-von-Guericke University, Universitätsplatz 2, 39106 Magdeburg, Germany
- 4 Research Centre in Digitalization and Intelligent Robotics (CeDRI), Instituto Politécnico de Bragança, Campus de Santa Apolónia, 5300-253 Bragança, Portugal
- 5 SMS Group GmbH, 40237 Düsseldorf, Germany

architectures (SOA) [2]. However, the integration of this collection of heterogeneous subsystems from different manufacturers into a functional unit still requires great engineering efforts [3].

For instance, complex chemical process plants are gradually transformed into a system of CPPSs. Currently, the overall system hierarchy follows the ISA 95 [4], with its clear separation of layers. Over time, additional services are added to the system. Integrating and maintaining these new services is a challenge. The services often span multiple layers of the original system, and flexible system architectures are needed without breaking compatibility with existing systems.

The general purpose of a system architecture in the terms of aPS is the connection of devices, tools and services, which together fulfil an aPS overall purpose. In this paper, the term *architecture* is defined as the connection of systems that enables sharing of data and services. Every component is referred to as a *participant*. Examples of such participants are human-machine interfaces (HMIs) or controllers. Field devices, e.g. sensors and actuators, are connected through participants to the architecture and never participate directly.

Recently, research activities have focused on the development of standardized system architectures for Industrie 4.0 (I4.0) which can overcome the current situation. While abstract reference architectures lack concrete realization strategies and details, numerous use-case specific proposals cannot be applied to other use-cases than the ones they were designed for. Therefore, there is a lack of generic architecture proposals for I4.0 that address the gap between reference and specific architectures.

This paper compares the requirements of three research projects in the field of technical system architectures for I4.0 and the specific architectures developed within these projects. Based on these, a generic system architecture is derived which captures a broader range of requirements and combines the strengths of the approaches.

This paper is structured as follows: first, the ongoing transformation process to I4.0 architectures is described. In the next section, related literature is discussed. Then, the projects involved are introduced and the requirements for a generic architecture are derived. A generic architecture is presented after an overview over project specific architectures. To evaluate this architecture, the section *Evaluation* reviews whether the generic architecture fulfills the requirements.

2 Migration paths towards Industrie 4.0

The realization of I4.0 principles in industrial automation challenges the architecture of the overall system. Classically, automation systems follow the organization pattern

of a layered architecture according to ISA 95 [4], separating the systems and their communication. This strong layering is a result of divergent requirements in the application domains. While the field layer, with its fieldbus communication, puts strict constraints on real-time behavior and determinism, the superordinate layers are characterized as Ethernet-based office networks [5]. Concepts that are aligned with I4.0, such as big data analytics [1] or self-awareness of CPPS [6], demand higher interconnectivity and harmonization of communication [7]. Thus, the monolithic ISA 95 system architecture interferes with these ideas and should be transformed.

However, the long lifetime of mission-critical systems results in many legacy systems that still need to be supported. Disruptive changes are not feasible and a gradual transformation is preferred. Secondary communication channels that enable data transfer across different layers, gateway concepts for the integration of legacy devices, and explicit semantics are possible contributions to this field. The steady evolution proposed can support operators on their migration path towards I4.0 and CPPS. Therefore, proven concepts for I4.0 system architectures are needed and can serve as a guideline to support operators in the migration process.

3 State-of-the-art in system architectures

Several standardization bodies, industrial consortia and research groups actively work in the field of system architectures for I4.0 to provide possible solutions for overcoming the layered structure with the aim of making system interaction more dynamic and flexible.

One of these initiatives is the German I4.0 initiative, which specified the Reference Architecture Model Industrie 4.0 (RAMI 4.0) [8]. RAMI 4.0 proposes an abstract reference model capturing the system hierarchy, the type of information represented, and the life cycle of assets. Moreover, the concept of I4.0 components is introduced. These are encapsulated inside an administration shell that is responsible for communication and includes orchestration and self-description mechanisms. Another abstract reference model is the American Industrial Internet Reference Architecture (IIRA) [9].

Lee et al. [10] define the 5C architecture for the realization of cyber-physical systems (CPS) with the levels smart connection (I), data-to-information conversion (II), cyber (III), cognition (IV), and configuration (V). The architecture serves as a guideline for implementations and realizations of CPS.

These models provide a technology-neutral starting point for I4.0 architectures and CPS; however, they lack a recommendation for how to realize such an architecture. Moreover, the support needed for legacy systems is only partly

considered. In parallel, several researchers have published architectures that provide more concrete insights into the technical realizations.

The ARUM project [11] proposes an agent-based architecture with an enterprise service bus (ESB) acting as middleware between the different systems. Legacy devices are incorporated using gateways. An ontology embedded in the middleware contributes to a common understanding of information.

Hufnagel and Vogel-Heuser [12] capture data integration from various heterogeneous sources using an ESB-based architecture. The architecture uses adapters to translate between data formats, thereby enabling the incorporation of legacy devices. A common information model with mapping rules that parametrize the data adapters serves to create a common understanding.

The Line Information System Architecture by Theorin et al. [13] uses an ESB for a prototypical implementation. The aim of the approach is to allow a flexible data integration in factories. A common information model and data adapters translate between the different systems.

The SOCRADES architecture [14] uses gateways and mediators for the integration of legacy devices. Web services facilitate interoperability and loose-coupling between the systems. Moreover, the discovery of services and their orchestration play an important role.

The Arrowhead project [15] provides a framework for the cloud-based interaction of systems. It closely follows SOA principles and considers data exchange across organizational borders. Additionally, it enables real-time capable communication if necessary.

Foehr et al. [16] compare a number of other recently published automation system architectures. The authors point out that most projects have developed aspects of an I4.0 architecture but lack a global view. Therefore, the authors point out the need for the integration of these separate aspects. Moreover, migration strategies are considered essential for industrial uptake of these technologies.

Summarized, numerous specific architectures for different applications exist. On the other hand, reference architectures provide a technology-neutral starting point for I4.0 architectures. However, still missing is a concrete system architecture, which bridges between reference and use-case specific architectures and provides an overall picture in enough detail.

4 Introduction to PERFoRM, IMPROVE and BaSys4.0

In the following, the two European projects PERFoRM and IMPROVE, as well as the German BaSys 4.0, are introduced in more detail.

4.1 PERFoRM: reconfigurability of aPS

The project Production harmonizEd Reconfiguration of Flexible Robots and Machinery (PERFoRM) targets the need for increasing flexibility and reconfigurability in manufacturing. Its main objective is to transform existing aPS into flexible and reconfigurable systems by providing an architecture with a common infrastructure for different industries [17, 18].

During the last few years, technologies and standards have been only partially implemented on the laboratory and testbed level. PERFoRM aims to consolidate these results and integrates them into an architecture for industrial automation that can be deployed into existing environments. The core of the project is to establish an adequate middleware, which links industrial field devices with upper IT systems. The outcomes are validated in a three phase process (development, implementation and test), starting with prototypes, then testbeds, and finally in real industrial scenarios.

The PERFoRM system is validated in four use cases, covering a wide spectrum of the European industrial force with diverse product complexities, production volumes, and processing types [19].

4.2 IMPROVE: virtual factory and data analysis

The innovative modelling approaches for production systems to raise validatable efficiency (IMPROVE) project aims to develop a decision support system for tasks such as diagnosis and optimization in aPS. This is realized by the creation of a virtual factory which serves as a basis for model development and validation. Therefore, data from several systems in the plant needs to be aggregated and integrated [20].

The use cases of IMPROVE are comprised of stretch-foil production, the production of composites, packaging lines for beverages, and the assembly of white goods. For example, the data that must be analyzed for foil production consists of specification data from engineering, historic operational data from an MES, and off-line quality data from a laboratory database. This data is scattered over multiple parties (OEM, client, service provider). An automatic data exchange between these parties is currently missing; therefore, the potential for combining operational and engineering knowledge for data analysis is lost.

4.3 BaSys4.0: runtime environment for I4.0 aPS

The project Basic System Industrie 4.0 (BaSys 4.0) develops a platform for the I4.0 age. Therefore, a virtual middleware and modularized digital twins are used to abstract the overall

production process and allow the optimization of the process before changing the plant configuration.

BaSys 4.0 includes seven demonstrators of different sizes and complexities. Besides lab demonstrators, industrial applications are in focus. One example is a simulation of a cold rolling mill for the production of aluminum coils, including a representation of superordinate systems and the communication between plant and these systems [21]. Moreover, real-time communication between the systems is considered.

5 Combining the strengths of the approaches

The architectures developed as part of IMPROVE, PERFoRM, and BaSys 4.0 are tailored to fulfill the respective project requirements but do not capture the overall picture. Combining these approaches and deriving a generic architecture has the potential to fill the gap between reference architectures and specific realizations. The requirements for the architecture from each project must be considered for this purpose. These requirements can then be consolidated to derive a generic architecture.

6 Requirements for Industrie 4.0 architectures

Next, the requirements from the projects are discussed.

6.1 General requirements: interoperability

All projects seek to enable easy integration and replacement of participants. Hence, every system architecture should satisfy flexibility as one of the major requirements.

A system architecture should be adoptable for different applications and use-cases of different sizes, which requires a scalable architecture. If the number of architecture participants increases due to the integration of additional tools, an architecture has to scale accordingly [22].

When a system is a combination of subsystems, the subsystems work together to fulfill an overall goal. The subsystems coordinate different sub processes of the manufacturing process or depend on the data generated in other systems. Replacing a participant should not lead to additional engineering effort being needed for adaptation. Hence, system architectures should integrate the participants modularly. The different participants should communicate with each other over standardized interfaces, reducing the necessity for point-to-point interfaces.

The interoperation of subsystems requires data to be processed which is sent and received by various subsystems.

These systems can stem from different manufacturers, leading to a heterogeneous combination of subsystems. A common information model which defines a common understanding of the data is required to allow communication between each subsystem and the architecture.

6.2 IMPROVE: data analytics and security

A major concern during data analysis in IMPROVE is data quality [23]. As common pre-treatment operations have to be carried out on the data, centralizing this functionality in a library minimizes overhead and the replication of code. Therefore, the architecture should provide centralized data curation services. In addition, embedded devices especially cannot provide historic data access due to resource constraints. Buffering streamed data on the architectural level for historic data access allows flexible data analysis.

In aPS, the data representing the overall process is scattered over distributed sources. These isolated data pools store partial views that do not reflect the overall context. To allow external partners access to joint analysis and specific data, a system architecture must support data sharing across organizational borders.

However, when sharing data, the sensitivity of data must be considered. For instance, the leaking of key performance indicators to an OEM only interested in the performance of specific equipment must be prevented. Thus, an architecture should ensure security, privacy and integrity of data. Access control, anonymization, and encryption must be supported by the architecture.

6.3 PERFoRM: service discovery

Besides common requirements like flexibility and scalability, PERFoRM introduces an additional requirement for service detection and orchestration. To support plug-and-produce processes, of registry and discovery mechanisms are considered. This enables automatic integration of heterogeneous hardware and software systems.

6.4 BaSys4.0: real-time capability

Deterministic real-time communication is required for participants on the field level and active control of the production process. Furthermore, it is important to separate real-time from non-real-time communication in order to ensure determinism.

6.5 Requirements for a generic system architecture

The requirements discussed are summarized and classified in Table 1. An X marks where a requirement is considered in the respective project. As can be seen from Table 1,

Table 1 Summary of architectural requirements

| Requirement | IMPROVE | PERFoRM | BaSys4.0 |
|--|---------|---------|----------|
| <i>Flexibility</i> Low-cost integration of subsystems. This includes the ability to react on changed requirements or new subsystems | X | X | X |
| <i>Scalability</i> No limitations through the size of an automation system or application domain | X | X | X |
| <i>Modularity and standardized interfaces</i> Replacing a system should not cause adaption of depending systems. Hence, the different subsystems should rely on standardized interfaces for modularity | X | X | X |
| <i>Common information model</i> Transparently processing data from different sources requires a common-information model | X | X | X |
| <i>Data curation</i> Data-based services build upon reliable data. Therefore, data curation services should be an architecture functionality | X | | |
| <i>Historic data access</i> Buffering streamed data for historic data access is a requirement in environments with constrained devices | X | | |
| <i>Inter-enterprise data exchange</i> Sharing data across parties lets enterprises work together | X | | |
| <i>Privacy, integrity and security</i> Sharing data with other systems is security critical. Hence, functionalities to ensure data privacy, integrity and security are necessary | X | | |
| <i>Service detection and orchestration</i> Seamless integration of participants. If new services are detected, these should be configured and used automatically | | X | |
| <i>Real-time communication</i> For distributed control components, support for real-time communication is required | | | X |

the projects overlap, having a certain number of common requirements. However, due to different use-cases and application domains, project-specific requirements exist as well. Thus, none of the projects fulfills all requirements listed, showing again the need for a generic I4.0 architecture.

7 Project-specific architectures

In the following section, the specific architectures from each of the three projects will be presented in more detail. These are designed to fulfill the project-specific requirements from Table 1.

7.1 PERFoRM: flexibility and reconfigurability

The PERFoRM architecture for seamless reconfiguration of production systems is based on a network of distributed systems, which expose their functionalities as services and are interconnected by an industrial middleware. The middleware ensures a transparent, secure, and reliable interconnection. Additionally it acts as a mediator between the communication partners, allowing systems which communicate using different protocols to interact with each other [24]. The PERFoRM architecture addresses all five levels of the 5C architecture by Lee et al. [10]. Figure 1 represents the overall PERFoRM architecture in which the participants can interact via standard interfaces.

The architecture functionality is not limited to the systems represented, but can be extended using the standard interfaces or technology adapters. Within the architecture, legacy tools, technology adapters, standard interfaces,

PERFoRM-compliant tools, and the middleware itself are foreseen as elements. The interfaces are standardized and describe how data and services can be accessed. Furthermore, an information model for the semantic description of data is specified. Systems which are developed directly within the project are expected to implement the standard interfaces.

Legacy tools are all components which are not developed directly within PERFoRM and therefore need to be adapted. Legacy tools can exist on the shop floor level, but also at the IT level. For each of these systems, specific technology adapters wrap and expose the legacy services in a PERFoRM compliant way [24].

7.2 IMPROVE: data exchange and data quality

The IMPROVE architecture [22] supports a multitude of different applications and tools. Standard interfaces are introduced to minimize the integration effort. In addition, this facilitates transparent data access and flexible reconfiguration. The layered architecture differentiates between data suppliers, data users, and dashboards. A common information model is embedded inside a middleware, the so-called data management and integration broker, through which interoperability between the participants can be ensured. Each connected system, therefore, has to support a subset of the overall model. Legacy systems, which are incompatible with the information model, are interfaced via data adapters. These translate transparently between representations of data. In addition, the middleware is able to provide data users with preprocessed data from other participants.

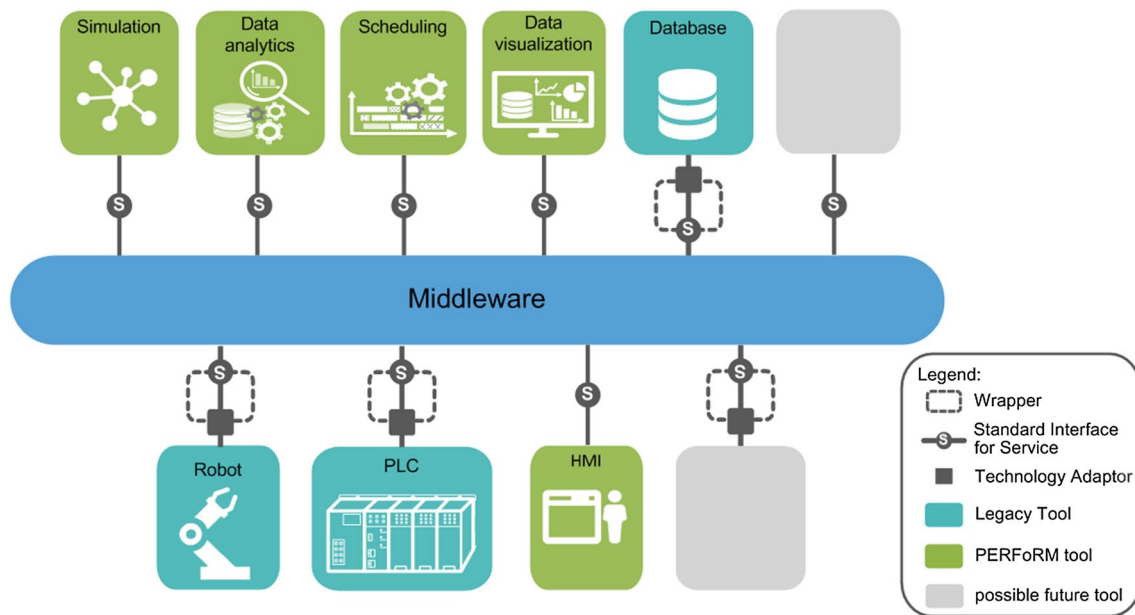


Fig. 1 Overall PERFoRM system architecture

Following a microservice approach, complex analysis processes can be divided into elementary steps.

Systems on the shop floor often lack the computational power and storage to provide historic data access, which implies that historic data should be stored in separate systems. Therefore, the architecture includes central data storage for historic data, analysis models, and results. Live data from the sources is streamed to this central repository. The data is available to all participants afterwards.

The data management broker curates data on-the-fly. As these are functionalities often needed by various analyzers, placing data curation on the broker level has the benefit of minimizing overhead. Depending on the use case, these functionalities could also be located in the analysis layer as separate microservices.

Furthermore, the broker includes an access control and anonymization layer. This layer verifies the access rights of the participants before any data is passed through the broker. Profiles ensure a granular differentiation of access rights. Additionally, an anonymizer component can normalize, introduce artificial noise, or mask metadata. Furthermore, encryption can be enforced to ensure security and data integrity. In the case of inter-enterprise connectivity, two independent brokers are connected over the internet and share data through a secure channel with an external data adapter on one side. The two-dimensional representation of the architecture in Fig. 2 shows an instance of the overall architecture for an organizational structure. Several of these instances can communicate with each other. The IMPROVE architecture addresses the levels I to IV of the 5C architecture by Lee et al. [10], with a stronger emphasis on levels

II (data-to-information conversion) and IV (cognition) than PERFoRM.

7.3 BaSys 4.0: real-time communication channel

An important aspect of BaSys 4.0 (see Fig. 3) is real-time communication between systems. The BaSys 4.0 middleware contains two distinct communication channels: a real-time enabled channel for time-critical applications and a non-real-time layer for less time-critical tasks. In this way, less critical communication does not interfere with control.

Real-time participants are connected to the architecture as systems which either fulfill their specific task (basic device) or combine more than one basic device in order to execute complex functions (group device). Examples of such devices are distributed controllers. Other participants are connected to the architecture on the plant level. These participants include, for instance, enterprise resource planning (ERP) systems and usually do not require real-time communication.

The middleware contains a directory service which itself stores addresses and protocol types of each participant, allowing communication using a standardized data format. The directory service is comprised of the functional structure as well as a description of the participants' functionalities. Data adapters allow diverse protocols for the connection of different systems.

Therefore, the focus of the BaSys 4.0 architecture is on levels I (smart connection) and III (cyber) of the 5C architecture by Lee et al. [10]. The architecture itself addresses the levels I–IV.

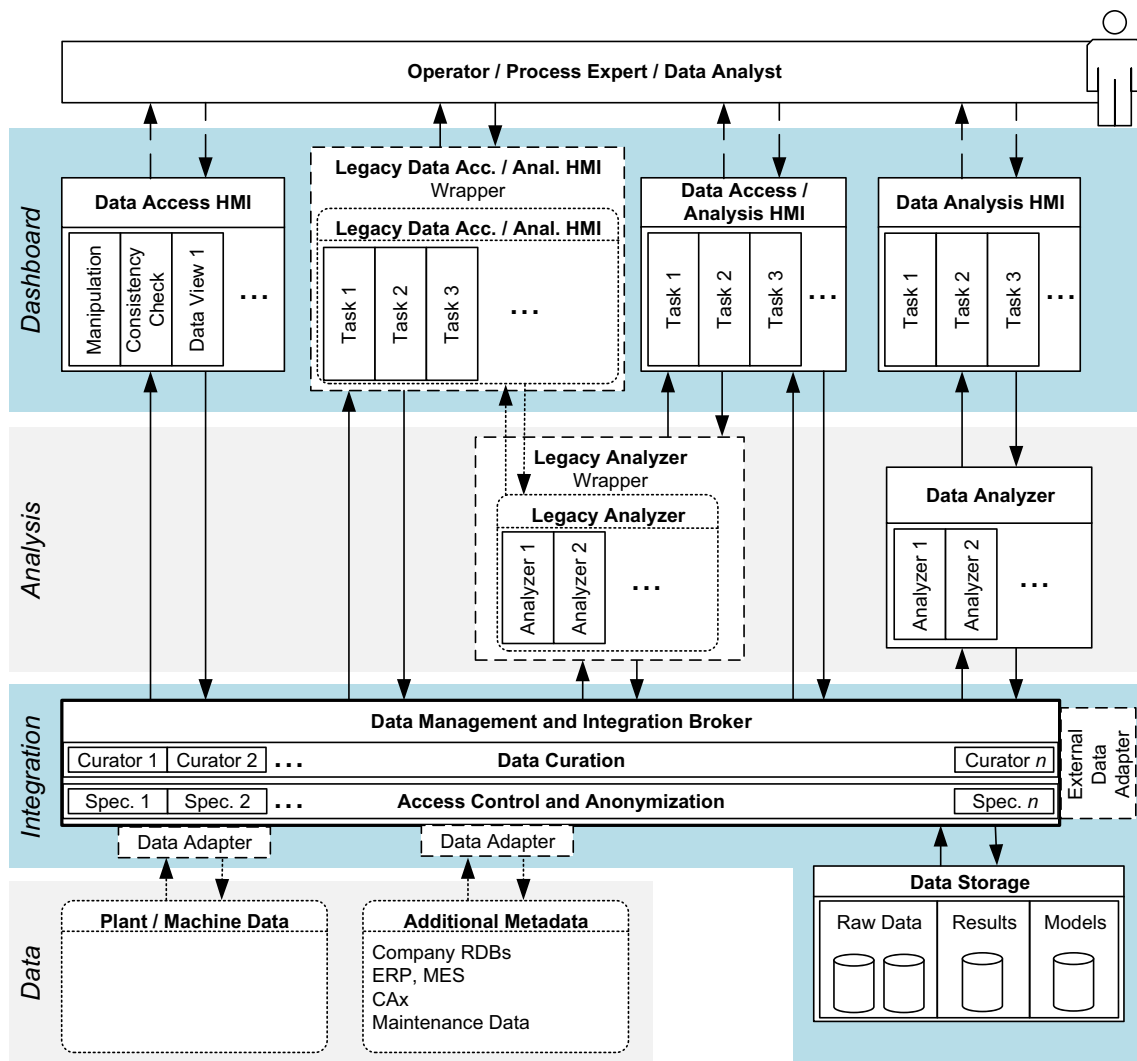


Fig. 2 Generic IMRPOVE architecture for data aggregation and integration of data from the shop floor

8 Derivation of the generic architecture

Based on the requirements identified from the projects and the specific architectures, a generic architecture which can serve as a basis for future developments is derived. The aim of the architecture is to address all five levels of the 5C architecture from smart connection up to configuration, but provide specific guidelines and examples for concrete realizations. For further implementation details, the reader is referred to the contributions cited in the project-specific sections.

The architecture’s heart is the data management and integration bus (cf. Fig. 4). This bus follows a middleware concepts for mediating between all connected systems, and contains two channels, a real-time and non-real-time channel. Participants requiring real-time data exchange can communicate through the real-time bus. The non-real-time bus

serves other participants. Only assigning time-critical data to the real-time bus saves communication costs and ensures proper real-time communication. Therefore, the bus is able to handle real-time as well as non-real-time communication. Possible technologies for the implementation of the bus are for instance Eclipse BaSyx Virtual Automation Bus [25], Data Distribution Service (DDS) [26], RabbitMQ [27], or OPC UA [28].

Data adapters translate between different information models and protocols. The PERFoRM information model (PML) [29] can serve as a basis for a common information model, which is implemented on top of AutomationML [30]. Depending on the use-case, other information models, such as DEXPI [31], could be employed. Adapters for legacy systems must be programmed individually for each information model and protocol. The adapters must contain information of the participants’ services to enable service detection (cf.

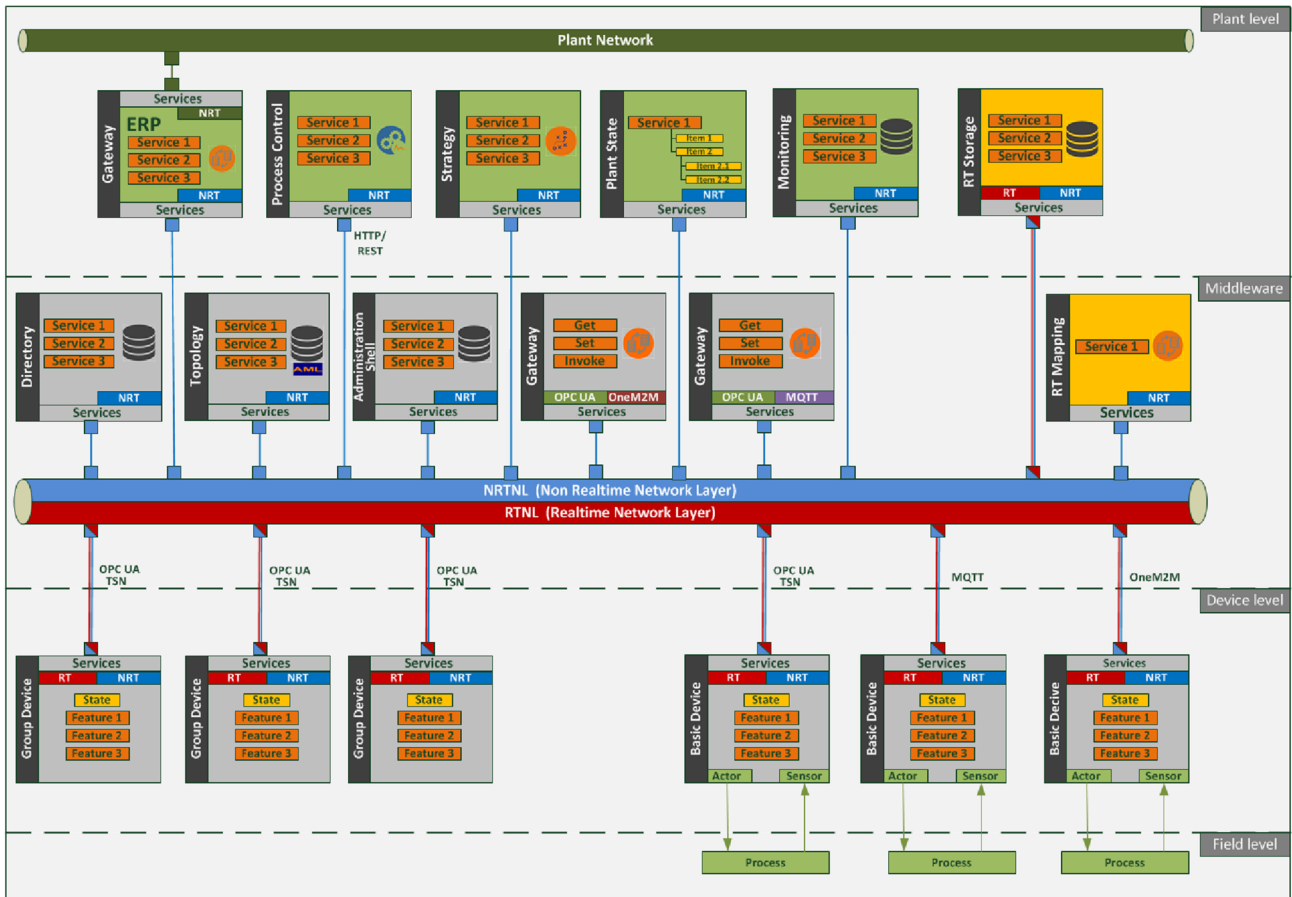
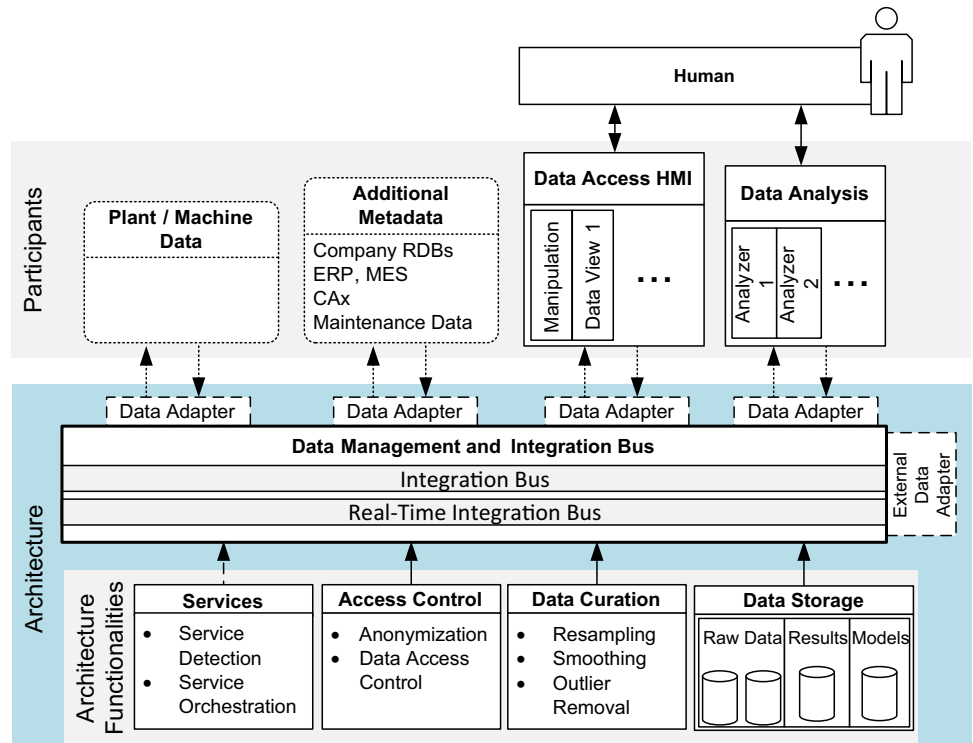


Fig. 3 BaSys 4.0 system architecture to enable non-real-time and real-time communication

Fig. 4 Generic architecture proposal based on the PER-FoRM, IMPROVE and BaSys 4.0 architectures



RAMI I4.0 components and the administration shell [8]). Similar to the data adapters for participants, an external data adapter serves as a translator for information models of other architecture instances used for inter-enterprise data exchange.

Additional functionalities can be enabled if required. For instance, an access control functionality can ensure data security through authentication (e.g. public-key authentication), encryption (e.g. symmetric encryption) and anonymization (e.g. normalization of data, introduction of artificial noise). Data curation functionality is crucial for data from participants exposed to stochastic effects, e.g. measurement noise, and can for instance remove outliers. Access control and data curation are embedded on the middleware layer to minimize redundant functionalities inside the architecture.

A data storage functionality automatically collects data from participants. Data is stored in a form that complies with the common information model. If the data storage itself is not compliant with the common information model, a data adapter can be used to translate between common and storage information models.

The service functionality detects services offered by participants. This is realized by using the service information received from the respective participant or data adapter. If two participants rely on each other's data, data orchestration can automatically build a link. Possible candidates for realization are OPC UA or DPWS [32]. The generic architecture derived from the requirements of the three projects is given in Fig. 4.

The generic architecture concept encompasses the intersecting aspects of the three presented project-specific architectures. Furthermore, it adds specific functionalities that can be enabled if required to fit various applications. The concept can also be mapped to other architectures published in literature (e.g. [11, 13]). Therefore it is aligned with the presented reference architectures, but provides a concrete and still generic enough starting point for I4.0 architectures.

9 Evaluation

In this section, aspects of the project architectures and the generic proposal are evaluated.

9.1 PERFoRM: application to use-cases

The use-case requirements were validated in pre-industrial testbeds before deployment in industrial use cases to de-risk the developed technologies in PERFoRM and to achieve a proof of concept. The core concept of the applied validation methodology describes test-scenarios reflecting the system and the use-case requirements, which will be refined to test cases. In particular, system and software testing based on

test scenarios and test cases is the state-of-the-art in the standard IEEE 29119 [33].

These test scenarios and technologies are discussed within the use cases in order to identify critical test scenarios. Critical test scenarios cover features which have to be de-risked in a test-environment before they can be implemented in a real industrial environment. In a last step, the tests are performed and the results documented. The validation and testing activities focus on the core elements of the PERFoRM middleware, standard interfaces, and adapter technologies. The general goal of the validation of the middleware solution is to test whether consumers are able to retrieve data in the common information representation PML which is routed through the middleware. Furthermore, the possibility of adding new participants by configuring new routes was tested. The evaluation demonstrated the suitability of the PERFoRM architecture in demonstration scenarios on a lab scale [34].

The interfaces between field level and middleware, as well as between middleware and superordinate IT systems, were successfully validated [35]. All use-cases use the same PML information model, which provides the exchange format for all participants. Adapter technologies were developed to enable the integration of legacy systems. For each use case, different legacy systems were considered.

The use-case providers perceive the results of the validation in testbeds positively. After the pre-industrial validation of the developed technologies in test beds, a next step is implementation in the use-cases.

9.2 IMPROVE: expert interviews and prototypical implementation

The IMPROVE architecture is evaluated in two different ways. On the one hand, expert interviews and questionnaires are used for determining the feasibility of the concept and the potential benefits. On the other hand, a prototypical implementation on a lab-scale is carried out as a proof of concept.

In [22], Trunzer et al. present the results of expert interviews. These interviews are carried out within the scope of IMPROVE and a second project, SIDAP [36]. SIDAP also puts a strong focus on data exchange across organizational borders. For the evaluation, the experts were questioned whether the concept can overcome the drawbacks of their current system layout. In addition, the experts were asked whether the implementation of the proposed architecture is feasible in their eyes.

During the interviews, the IMPROVE architecture is positively evaluated by the experts. The shortcomings of the current system layout (data integration by hand, lack of data understanding, no automatic data acquisition) are potentially solved. The experts point out the need for a common

information model in order to enable automatic processing of data. Furthermore, the standardized interfaces and defined means of cross-enterprise data sharing with access control are seen as advantages. The effort required for initial deployment inside a company is identified as a potential problem. Therefore, a parallel deployment is proposed, migrating system by system while leaving the hierarchical structure untouched at first and establishing a secondary communication channel.

For a proof of concept, the architecture is prototypically implemented on a lab-scale [27]. Therefore, a programmable logic controller is connected via a data adapter to the middleware, in this case RabbitMQ. Additional data sources are connected to the broker via data adapters. On the middleware level, a common information model is used. An access control and anonymization layer with preconfigured rules handles requests to the architecture. Furthermore, data is anonymized based on access rights. Data storage buffers live data for historic access. Two analyzers analyze the data in order to reveal hidden knowledge. While one of the analyzers works solely on streamed data from the plant, the other analyzer uses historical data in parallel. The architecture simplifies access to data greatly. Moreover, data can easily be exchanged between the systems. A granular management of access rights is possible due to the embedded access control and anonymization layer. Usage and adaption of well-accepted technologies from other domains, e.g. AMQP brokers such as RabbitMQ for the middleware, saves costs and minimizes development time.

9.3 Generic architecture for the factory of the future

The generic architecture for the Factory of the Future combines the approaches of the project-specific architectures presented here. Therefore, besides the requirements that are common to all three architectures, it also has to fulfill the specific aspects. These additional functionalities are considered in the generic architecture concept as architecture functionalities that can be adapted and enabled if required. Different functionalities are enabled when being applied, while the core of the generic architecture is the same.

Therefore, the generic architecture is, by design, capable of being applied in specific use-cases. The evaluation of the different aspects carried out in the projects also holds true for the combined, generic architecture. Through the evaluations carried out in each project, the subfunctionalities under consideration have been successfully verified. The feasibility and applicability of the basic middleware concept and the data adapters has been demonstrated for several use-cases [22, 27, 35]. Additionally, with PML [29, 35] a candidate for a common information model and its integration into the architecture has been successfully evaluated. The additional

functionalities have been evaluated on top of the project-specific architectures. As the generic architecture concept comprises the architecture concepts common to all three approaches, these functionalities can easily be integrated into the generic architecture.

10 Conclusion and outlook

The increasing heterogeneity and complexity of production lines has brought classical aPS architectures to their limits. Interconnectivity and flexibility have become more important as more logic is embedded into CPPS. New system architectures are necessary to serve these requirements. Most importantly, a gradual evolution of the monolithic automation pyramid into flexible I4.0 architectures is needed in order to maintain support for existing legacy systems.

Several approaches exist that enable interoperability; however, most address a specific field of application. Thus, many isolated views with a lack of synchronization between them can be found. In this contribution, the authors derive a generic, widely applicable architecture combining the strengths of the isolated approaches. This generic architecture proposal bridges the gap between reference and use-case specific architectures.

The generic architecture features a middleware with data adapters for interoperability. Data adapters allow communication of the architecture with other distributed systems. A separate real-time communication layer allows time-critical communication. Additional architecture functionalities, such as data curation, service orchestration, and access control are organized in a library and can be enabled on demand.

Application of the generic architecture inside projects is considered as future work. Through this, the applicability in the use-cases can be evaluated. Furthermore, the authors want to encourage other research groups to use and adapt the proposed architecture concept. Comparative studies on alternative realization concepts and technologies for implementation can further be of interest. The standardization of I4.0 architectures is of special interest for the future. Besides an alignment of national initiatives and a unification of efforts, guidelines and realizations should be developed. The first initiatives for joining forces have already been taken in the G20 [37].

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Vogel-Heuser B, Hess D (2016) Guest editorial industry 4.0—prerequisites and visions. *IEEE T-ASE* 13(2):411–413
2. Leitão P, Colombo AW, Karnouskos S (2016) Industrial automation based on cyber-physical systems technologies: prototype implementations and challenges. *Comput Ind* 81:11–25
3. Vogel-Heuser B, Diedrich C, Pantforder D, Göhner P (2014) Coupling heterogeneous production systems by a multi-agent based cyber-physical production system. In: 12th IEEE INDIN, IEEE, pp 713–719
4. The Instrumentation, Systems, and Automation Society (2000) Enterprise-control system integration—part I: Models and terminology (ANSI/ISA-95.00.01)
5. Sauter T (2007) The continuing evolution of integration in manufacturing automation. *IEEE Ind Electron Mag* 1(1):10–19
6. Lee J, Kao HA, Yang S (2014) Service innovation and smart analytics for industry 4.0 and big data environment. *Procedia CIRP* 16:3–8
7. Wollschlaeger M, Sauter T, Jasperneite J (2017) The future of industrial communication: automation networks in the era of the internet of things and Industry 4.0. *IEEE Ind Electron Mag* 11(1):17–27
8. Deutsches Institut für Normung eV (2016) Reference architecture model industrie 4.0 (RAMI4.0) (DIN SPEC 91345)
9. Industrial Internet Consortium (2017) The industrial internet of things. volume G1: reference architecture. https://www.iiconsortium.org/IIC_PUB_G1_V1.80_2017-01-31.pdf
10. Lee J, Bagheri B, Kao HA (2015) A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manuf Lett* 3:18–23
11. Leitão P, Barbosa J, Vrba P, Skobelev P, Tsarev A, Kazanskaia D (2013) Multi-agent system approach for the strategic planning in ramp-up production of small lots. In: *IEEE SMC2019*, IEEE, pp 4743–4748
12. Hufnagel J, Vogel-Heuser B (2015) Data integration in manufacturing industry: model-based integration of data distributed from ERP to PLC. In: *IEEE 13th INDIN*, pp 275–281
13. Theorin A, Bengtsson K, Provost J, Lieder M, Johnsson C, Lundholm T, Lennartson B (2016) An event-driven manufacturing information system architecture for industry 4.0. *Int J Prod Res* 55(5):1297–1311
14. Karnouskos S, Bangemann T, Diedrich C (2009) Integration of legacy devices in the future SOA-based factory. *IFAC Proc Vol* 42(4):2113–2118
15. Delsing J (2017) *IoT automation: arrowhead framework*. CRC Press, Taylor & Francis Group, Boca Raton
16. Foehr M, Vollmar J, Calà A, Leitão P, Karnouskos S, Colombo AW (2017) Engineering of next generation cyber-physical automation system architectures. In: Biffl S, Lüder A, Gerhard D (eds) *Multi-disciplinary engineering for cyber-physical production systems: data models and software solutions for handling complex engineering projects*. Springer, Cham, pp 185–206
17. PERFoRM Consortium (2019) PERFoRM. <http://www.horizon2020-perform.eu>. Accessed 23 Apr 2019
18. Leitão P, Barbosa J, Foehr M, Calà A, Perlo P, Iuzzolino G, Petrali P, Vallhagen J, Colombo AW (2017) Instantiating the PERFoRM system architecture for industrial case studies. In: Borangiu T, Trentesaux D, Thomas A, Leitão P, Oliveira JB (eds) *Service orientation in holonic and multi-agent manufacturing, studies in computational intelligence*, vol 694. Springer, Cham, pp 359–372
19. Calà A, Foehr M, Rohrmus D, Weinert N, Meyer O, Taisch M, Boschi F, Fantini PM, Perlo P, Petrali P, Vallhagen J (2016) Towards industrial exploitation of innovative and harmonized production systems. In: 42th IECON, IEEE, pp 5735–5740
20. IMPROVE Consortium (2019) IMPROVE. <http://www.improve-vfof.eu>. Accessed 23 Apr 2019
21. Terzimehic T, Wenger M, Zoitl A, Bayha A, Becker K, Müller T, Schauerte H (2017) Towards an industry 4.0 compliant control software architecture using IEC 61499 & OPC UA. In: 2017 IEEE ETFA
22. Trunzer E, Kirchen I, Folmer J, Koltun G, Vogel-Heuser B (2017) A flexible architecture for data mining from heterogeneous data sources in automated production systems. In: 2017 IEEE ICIT, IEEE, pp 1106–1111
23. Kirchen I, Schütz D, Folmer J, Vogel-Heuser B (2017) Metrics for the evaluation of data quality of signal data in industrial processes. In: *IEEE 15th INDIN*, IEEE, pp 819–826
24. Gosewehr F, Wernann J, Borysch W, Colombo AW (2017) Specification and design of an industrial manufacturing middleware. In: *IEEE 15th INDIN*, pp 1160–1166
25. The Eclipse Foundation (2019) Eclipse BaSyx. <https://www.eclipse.org/basyx/>. Accessed 23 Apr 2019
26. Object Management Group (OMG) (2015) Data distribution service (DDS). <https://www.omg.org/spec/DDS/1.4/PDF>
27. Trunzer E, Lötzerich S, Vogel-Heuser B (2018) Concept and implementation of a software architecture for unifying data transfer in automated production systems. In: Niggemann O, Schüller P (eds) *IMPROVE—innovative modelling approaches for production systems to raise validatable efficiency: intelligent methods for the factory of the future*. Springer Berlin Heidelberg, Berlin, pp 1–17
28. International Electrotechnical Commission (2016) OPC unified architecture—part 1: Overview and concepts (IEC 62541-1)
29. Angione G, Barbosa J, Gosewehr F, Leitão P, Massa D, Matos J, Peres RS, Rocha AD, Wermann J (2017) Integration and deployment of a distributed and pluggable industrial architecture for the PERFoRM project. *Procedia Manuf* 11:896–904
30. International Electrotechnical Commission (2018) Engineering data exchange format for use in industrial automation systems engineering—automation markup language—part 1: Architecture and general requirements (IEC 62714-1)
31. Wiedau M, von Wedel L, Temmen H, Welke R, Papakonstantinou N (2019) Enpro data integration: extending DEXPI towards the asset lifecycle. *Chemie Ingenieur Technik* 91(3):240–255
32. Leitao P, Karnouskos S, Ribeiro L, Lee J, Strasser T, Colombo AW (2016) Smart agents in industrial cyber-physical systems. *Proc IEEE* 104(5):1086–1101
33. International Organization for Standardization (2016) Software and systems engineering—software testing—part 5: Keyword-driven testing (ISO/IEC/IEEE 29119-5)
34. Chakravorti N, Dimanidou E, Angione G, Wermann J, Gosewehr F (2017) Validation of PERFoRM reference architecture demonstrating an automatic robot reconfiguration application. In: *IEEE 15th INDIN*, IEEE, pp 1167–1172
35. Dias J, Vallhagen J, Barbosa J, Leitao P (2017) Agent-based reconfiguration in a micro-flow production cell. In: *IEEE 15th INDIN*, IEEE, pp 1123–1128
36. SIDAP Consortium (2019) SIDAP. <http://www.sidap.de>. Accessed 23 Apr 2019
37. Federal Ministry for Economic Affairs and Energy (2017) *Digitising manufacturing in the G20—initiatives, best practice and policy approaches: conference report*