PLC 4.0: A Control System for Industry 4.0

Mahyar Azarmipour[†], Haitham Elfaham[†], Caspar Gries[§] and Ulrich Epple[†]
[†] Chair of Process Control Engineering
RWTH Aachen University, Aachen, Germany
Email: {m.azarmipour, h.elfaham, epple}@plt.rwth-aachen.de
[§]SYSGO GmbH, Klein-Winterheim, Germany
Email: caspar.gries@sysgo.com

***Abstract*—Industry 4.0 follows the goal of promoting digitalization of manufacturing. Thereby converging information technologies and automation play a crucial role. These goals do not only require new softwares and information models but also require hardwares that offer more intelligence and bring more agility in production systems. The current developments in field of sensors introducing sensor 4.0 is an example of adapting the hardwares to industry 4.0 requirements. This is also necessary for other participating hardwares such as control devices. The control devices must be also designed to act more intelligent and provide interfaces to the other technologies like cloud computing and Cyber Physical Systems. In this paper, a new architecture for control devices will be introduced that operates as a bridge between industrial automation applications and IT technologies and provides a virtual platform for tests and simulations that run parallel to the control procedure. We refer to this architecture as "PLC 4.0".**

***Index Terms*—Industry 4.0, PLC 4.0, Information Technologies, Virtualization, Cloud Computing, Cyber Physical System**

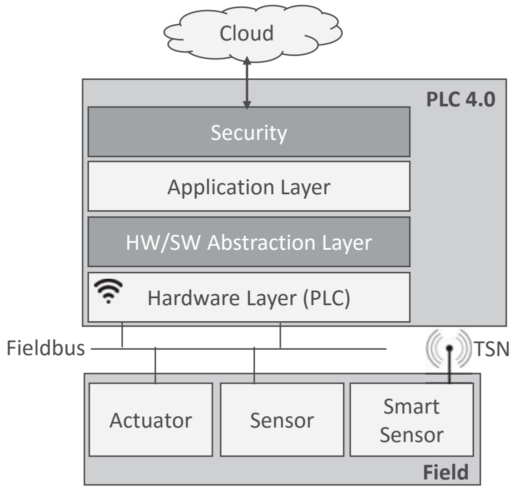

Fig. 1. Infrastructure of PLC 4.0

I. Introduction

Recently, in the industrial automation domain, substantial paradigm shifts are taking place. Initiatives like Industry 4.0 (I40) in Germany, Industrial Internet Consortium (IIC) in USA and many others aim to introduce revolutionary ideas and concepts like Internet of Things (IoT) [1], cloud computing, Cyber Physical Production System (CPPS) to achieve objectives like adaptability [2]. Whereby adaptability describes the readiness of a system to react to unplanned changes or modifications [3]. Establishing these concepts requires the modification of fundamental aspects in hardware and infrastructure (smart field devices), increasing the interconnectedness through dissolving the communication hierarchy in the automation pyramids or other development aspects like service orientation, interoperability, virtualization of deployment, P2P communications, etc. In the traditional automation systems, mass production and fast manufacturing are in focus. Whereby the flexibility of the automation systems is based on recipes. I4.0 follows the goal of promoting digitalization of manufacturing. Future automation systems allow a digital engineering of products and production processes. Thereby integrating Information Technologies (IT), such as cloud computing, IoT and CPPS in automation play a crucial role to increase the intelligence of the manufacturing systems. Current developments can be witnessed in the field of sensors introducing sensor 4.0 or "smart sensors". However, further development can be done in the infrastructure namely in the control layer. This paper deals with a new architecture for PLCs. We refer to it as "PLC 4.0". The suggested architecture provides a platform that enables connectivity with public cloud systems while also considering security aspects. Furthermore, it provides an abstraction layer that increases the agility of the PLC and enhances the communication Quality of Service (QoS) and scalability. Fig. 1 illustrates the suggested infrastructure of the PLC 4.0. The dark gray boxes demonstrate the introduced layers that can provide the aforementioned enhancements.

The remainder of this paper is structured as follows: Sec. II deals with related works as well as basic knowledge and technologies that are used by the concept. Sec. III discusses the fundamental features of PLC 4.0. The concept and related scenarios are presented in Sec. IV and V respectively. The implementation is discussed in the Sec. VI. Finally, Sec. VII concludes the paper.

II. Related Works and Fundamentals

Recently, some approaches to implement virtual PLCs (vPLCs) in order to replace hardware PLCs are introduced.

The authors were supported by German Federal Ministry of Education and Research in the scope of BaSys 4.0 project (Foerderkennzeichen 01jS16022).

978-1-7281-4878-6/19/$31.00 ©2019 IEEE

For example, [4] introduces an approach to offer control as a service in industrial automation to increase agility of automation systems. This approach implements PLC as a virtual entity in a private cloud (to ensure security) that is delivered to field devices as a service. The introduced approach in [7] follows a similar goal using AMAZON services. The proposed method in this work deals with a new architecture for hardware PLCs that offers also cloud communication (public or private) considering security and introduces some methods to increase the agility of automation systems. In [6], an architecture for a multi-purpose controller using container technologies is introduced. This paper applies both container technologies and hypervisor solutions to ensure dynamic deployment and connectivity as well as security, separation of critical applications, real time Operating Systems (OS) and communication.

A. Virtualization

In I40, virtualization plays an important role in the system design. Virtualization can provide an abstraction layer for software increasing the agility of the system. Furthermore, virtualization provides better control to the resources distribution amongst the different applications/operating systems. The different virtualization schemes are explained in the following sections:

a) Virtualization by means of a Hypervisor: The concept of Virtual Machines (VMs) and virtualization by means of a hypervisor was introduced by IBM [18], [19]. A hypervisor is an OS for OSs. It is a software layer that abstracts the low level platform interface such as processor, memory and network connectivity to allow applications to run simultaneously [5]. These applications, their OS and all other dependencies are encapsulated in isolated environments called VM. There are two types of hypervisors. Type 1 hypervisor can be installed directly on a hardware without the need for an OS (see Fig. 2(a)), while type 2 hypervisors run as regular applications within an OS (see Fig. 2(b)).

b) Virtualization by means of a container: Container technologies can also be considered as a virtualization method. Similar to a VM, a container allows abstraction, however, only at the application level. Since applications running in containers make use of the underlying OS, they are light-weight in comparison to VMs, which bundle the application with a dedicated OS instance (see Fig. 2(c)). Container technologies offer a solution for decentralization, scalability and dynamic deployment.

B. Dynamic Runtime Environments

Dynamic Run-Time Environments (RTE) in industrial automation provide a platform for logic execution that can be dynamically adjusted at runtime. The RTE utilized in our example is the ACPLT/RTE. In the library development stage (offline stage), the logic of meta models and classes for example function blocks, control charts or sequential charts is written in ANSI/C and consequently compiled producing .dll or .so files according to the OS. These meta models

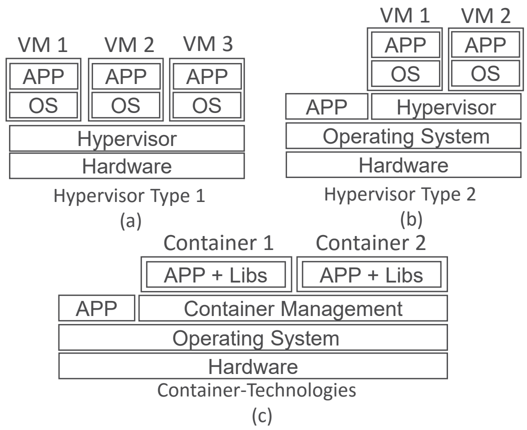

Fig. 2. Virtualization methods

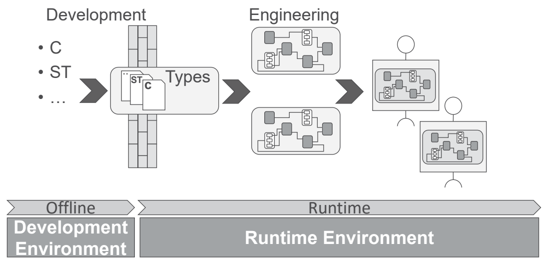

Fig. 3. Dynamic Runtime Environments

and function block types can be instantiated and engineered (connected with one another) at runtime. Library exchange between compatible OS is also possible. Fig. 3 demonstrates the different phases of the RTE.

C. Automation Hierarchy

The automation pyramid classifies the automation hierarchy to four layers namely (in a bottom up order): field, control, MES and ERP. Each layer is responsible for different objectives. In the classic automation paradigm, the communication infrastructure is built in a coherent manner. With the proposition of dissolving the communication hierarchy in I40, new objectives as well as risks arise. One of the main concerns is security namely at the control layer. Having open communication endpoints for the control layer can induce the risk of external attacks or manipulation of control software. In the proposed PLC 4.0 scheme, these concerns are addressed.

1) Control Hierarchy: In industrial automation, control logic is implemented in a specific hierarchy that ensures an orchestrated functionalities between the different components as shown in [16]. This control hierarchy consists of Single Control Units (SCU), Group Control Units (GCU) and Procedures. SCUs contain the control logic of the individual field devices. GCUs orchestrate the different functionalities of the SCUs altogether. This hierarchy has a flexible scheme based on procedures that can be instantiated and executed using aforementioned control units as shown in Fig. 4.

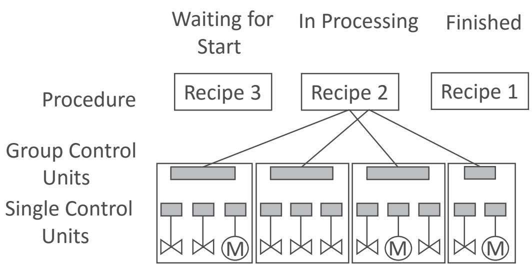

Fig. 4. Control Hierarchy

D. Digital Twin

The term Digital Twin (DT) was introduced by Grieves in 2003 in University of Michigan. However, the definition of the term DT evolved and has been introduced in different contexts as shown in [20]. Despite the various definitions of a DT, one common objective of DT is to virtualize the different aspects of a system according to use-case in the information world. The DT technology is usually defined through three parts. Physical space, virtual space and the connection of two spaces [10]. This fulfills one of the main goals in context of industry 4.0, namely digitalization by mapping the physical production systems in digital world to test their behavior. It does not only contain the model of the asset but also their interactions and interfaces, as well as their manufacturing processes [17]. This can be used as a basis for data analyses for optimization, validation, fault diagnosis etc.

E. Asset Administration Shell, Industry 4.0 Component

Asset Administration Shell (AAS) is a digital representation of an asset during its life cycle. The objective of AAS is to provide consistent information about the asset. For example, an AAS of a drill contains information about manufacturer, rotation speed, drilling depth etc. Uploading an AAS to the cloud is a common practice as it provides global access and also ensures its availability despite the asset connectivity.

III. Fundamental Features for PLC 4.0

A. Cloud Solutions - External Interface

Integrating IT solutions in automation is a pursued objective in I4.0. In this regard, integrating cloud solutions in automation increases the adaptability of automation system by allowing a dynamic deployment of applications [8]. A realization of an app-store [13] like concept in automation builds a dynamic system of applications as a basis for an adaptable production. These applications (and their required compiled libraries) can be thus downloaded and executed in runtime provided that the runtime environment provides such dynamic integration of libraries and applications (cf. Sec. II). Furthermore, concepts like AAS should be safely and seamlessly integrated in the new automation architecture. Hence, an external interface is required which can be provided within a container. This container will have restricted rights (in terms of communication and resources consumption) and will act as a portal to the external cloud systems.

B. Simulations and Validations

Simulation is an important factor that must be considered and modified to meet the I4.0 requirements as mentioned in Sec. II-D. Bridging reality and virtual simulation, e.g., through digital twins, corresponds to adopting information from the virtual world to the real world i.e., in our case, the running implementation. Scenarios like fast forward predictions, that can be used by the operator as an assistance to detect system failures and problems prior to occurrence through plant simulations, can be implemented. Furthermore, the aforementioned cloud solutions allow a dynamic deployment of applications but these applications must be tested for validity prior to be used in operation. These simulations and validations should not compromise the system safety and its resources management.

C. Realtime and QoS Assurances

Real time execution and QoS (communication latency and safety) can be administered using VM as shown in Hypervisors of Type 1 in Sec. II-A. Furthermore, software abstraction can allow the execution of RTOS (Real-Time Operating Systems) ensuring the real time requirements while simultaneously running other VMs that handle other tasks (e.g. external interface).

D. Adaptability

Adaptability on the control hardware level includes features such as:

- Scalability: in order to cope with the various available operating systems and increasing dependencies by the applications, different computation hardware are utilized parallely. Different applications with different dependencies can share the same hardware instead of running on different hardware resources using hypervisors and VMs. Virtualization is a key technology to provide highly scalable resources to reduce under-utilized resources [14]
- Deployment: undisrupted software updates and deploying new applications
- Resources management: distributing the available resources amongst the running applications according to a defined adjustable scheme (priority, criticality, etc.)

IV. PLC 4.0 Concept

Introducing a new architecture for the control device can serve as a solution to provide a seamless integration for the aforementioned features without compromising aspects like security, safety, etc. The introduced architecture uses virtualization technology and provides a strict isolation of physical and virtual spaces maintaining security and allowing for independent execution. Therefore, updating, resetting or modifying a partition can be done securely without affecting the operation of other running containers. Moreover, it provides the possibility of dynamically reallocating the resources according to the requirements. Two important aspects in this system are the real time communication between the partitions and applications and the system security. Security deals with the external (e.g., cloud) and internal (e.g., another partition)

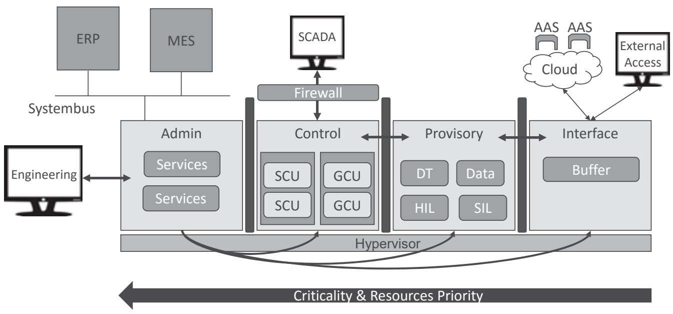

Fig. 5. PLC 4.0

access to the different partitions. This can be defined and can also be limited to authorized persons.

In the introduced concept, the partitions are classified into three meta classes; Control, Provisory and Interface. Fig 5 demonstrates the architecture showing the different partitions and the communication flow between the different entities. Each meta class has a defined set of rules regarding for example QoS, communication, resources consumption, etc. These rules are defined by the Admin partition. The Admin meta level comprises only one partition.

A. Admin

Admin controls and monitors all functions and operations of PLC 4.0. It is connected to higher automation layers such as ERP and MES to adapt the operation of PLC to the system requirements or changes related to failure, safety etc. This unit offers different services for the management of PLC 4.0. These services can be categorized as follow:

- CRUD Services: It deals with services like create, read, update and delete that are offered for component management.
  - Create: This service creates a component and assigns required storage and computing resources to it.
  - Read: This service reads all existing components and their used resources as well as the existing communication channels.
  - Update: This service is used to update components.
  - Delete: This service deletes a components or a communication link between them.
- Services for Resource Allocation: Admin monitors the resource allocation and is able to reallocate the resources according to the criticality of applications.
- Services for Communication: The communication between components in different partitions is monitored and controlled by Admin. It offers different Quality of Services (QoS) to secure a reliable communication between components.
- Domain Specific Services: These services are offered according to the use-case.

1) Design Methods: For the realization of Admin, two different design methods are considered. These designs should allow a modularity in the implementation of services so that they can be individually managed and deployed.

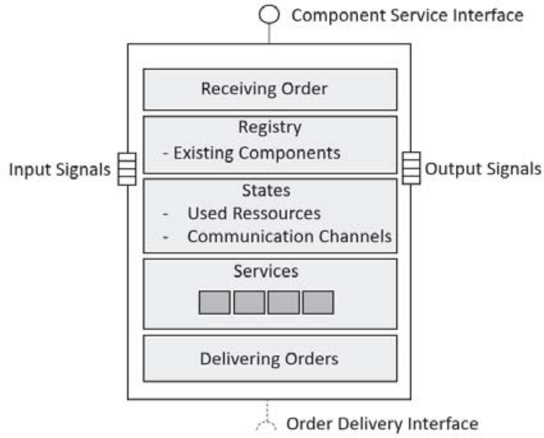

Fig. 6. Manager Component

a) Microservice-Architecture: Microservices are a variant of the Service-Oriented Architecture (SOA), in which the applications are decomposed into smaller services. This decomposition allows more modularity and granularity. These fine-grained services can be implemented in different languages. The modularity and loose coupling of services allows deployment of services. The services can be deployed or delivered according to the required skills. This services can be delivered by higher automation layers such as MES or ERP. This implementation leads to an application that is composed of loosely coupled services [21].

b) Component Based Architecture: Component based architecture increases the modularity of system by generating autonomous units that can be reused and deployed. In this case the manager is implemented as a component that offers its services over its interfaces. The Admin is a component and can be also managed as a component. This means it can be updated, deleted, etc. by an operator or a higher automation layer such as MES. Fig. 6 represents the internal structure of such a component. It has a registry unit in which all existing components are registered. Different hardware states such as used resources or communication channels are represented. The last layer represents the offered services.

B. Control

Control meta class contains the control procedure of the system that is based on the control hierarchy defined in Sec. II. It consists of SCUs and GCUs that are executed in a dynamic runtime environment and therefore can be changed during run time. It offers different interfaces for SCADA, Admin and also the operator to revise or update the control logic.

C. Provisory

Adaptability is one of the main goals of I4.0. An adaptable system must be able to change its behavior to meet the current requirements. Thus, introducing a provisory meta class which acts as a bridge between simulated models (e.g. Software and

Hardware in the loop) and implementations can seamlessly integrate the digital world in the automation infrastructure. This unit offers the possibility of testing and validating the new applications before being applied in control system. Moreover, it is also a platform for a parallel simulation of the running control logic as a basis for fault diagnosis, optimization etc.

D. Interface

As mentioned before, in context of I4.0 the strict separation of automation pyramid layers is replaced by an interconnected hierarchy. Integrating an interface to a technology like cloud computing in the control device is the basis for a high connectivity and dynamic deployment. The interface meta class provides a portal to the cloud and contains the only partition that can communicate with the outside world to download and upload required components.

V. Scenarios

A. A platform for Digital Twin

DT builds a virtual model of involved physical entities, in order to simulate their behavior [9]. This virtualization is a basis for optimization of the production, machine learning, fault diagnosis, etc. It consists of a physical and virtual space as well as a real time communication between them. Thereby the physical and virtual space are totally independent from each other. DT can be investigate from different perspectives. One of these perspectives is related to framework and infrastructure. Framework is studied in [11] based on CPPS and in [12] based on cloud computing. In [12], an architectural framework for DT is introduced but an infrastructure for the interaction between virtual and physical space is not addressed.

The proposed control architecture in this paper provides an infrastructure for digital twin, in which physical (control) and virtual (provisory) space are absolutely separated from each other and at the same time can communicate in real time. The virtual space contains DT, plant models and data that are needed for the simulation and physical space contains the control logic. In this case the DT acts on control level to predict, optimize and fault diagnose of the control procedure. In this infrastructure, Admin can control all the interactions between physical and virtual space. The benefits of using this infrastructure are as follows:

1) Offering real time operating systems and real time communication between Partitions: Hypervisor (depending on technology) offers a real time communication between partitions eliminating the need to use another technology in order to realize a real time communication.

2) Security: As mentioned before the communication can be considered from two perspectives. These perspective are the access to the VMs and the communication between them. The VMs are fully independent and separated from each other. The access to the VMs can be limited to the authorized persons. The communication between VMs can be controlled by manager unit. The manager can decide if two components are allowed to communicate or not. The communication can be bidirectional or one directional, and can be specified by rights such as Read, Write, copy etc.

B. Adaptability

One of the main goals of I4.0 is adaptability. This indicates the readiness of the production system to react to unplanned changes. An example of such a system is a system that is designed to produce variants of a product. Adapting the production in this case requires manipulating the control logic. In order to prevent an interrupt in the production, this should be done during run time. The proposed infrastructure offers a platform for updating the control logic or its OS.

1) Adaptability and Dynamic Change between Active control: The provisory partition in Fig. 5 offers a platform for making changes in control logic or its softwares and OSs. The control logic can be modified, updated or optimized in order to meet the new changes and conditions in production system. These necessary changes can be made while the active process control runs unaffected forward. Thus the provisory and active process control can exchange their role so that the updated control logic overtakes the handover of the production. In this case the control and provisory partition should be assigned with the same rights and security level, since they both contain the control procedure of the production system.

2) Synchronization: Switching active process control and Provisory requires a synchronization between control procedure so that the process control continues without interruption. In the next research steps, the possibilities of synchronizing these procedures will be taken into consideration.

VI. Prototype Realization

In this Section, the implementation of this virtualized system is discussed. For the implementation of device level a type 1 hypervisor is used. We use PikeOS hypervisor solution of SYSGO GmbH. With the help of the hypervisor, real time communication between partitions by means of FIFO Queue can be established. In the different partitions, we use the runtime environment ACPLT/RTE ported to the real time OS of PikeOS and the communication between function blocks takes place by means of FIFO Queue (Real time). Furthermore, we use Docker [15] as the Container technology to perform the deployment in the different partitions.

Fig. 7 represents the plant used in our prototype. The plant regulates the levels of the two comprised tanks. Each tank is equipped by a pump and flow controlled valve which that control its level. Additionally, the tanks are equipped with a level sensor and two stages overflow prevention mechanism; using two cascaded valves (one flow controlled valve and another pump-bypass valve) and an emergency overflowing pipe respectively. In the considered use-case, a uni-directional flow from Tank 1 to Tank 2 is allowed. The objective is to control the level at Tank 2 by the means of newly introduced strategy that should replace the available one. Using the proposed PLC 4.0 architecture, we demonstrate a safe testing and eventually a smooth transition of control by toggling the control strategies. The PLC 4.0 concept for this plant is represented in Fig. 7. It

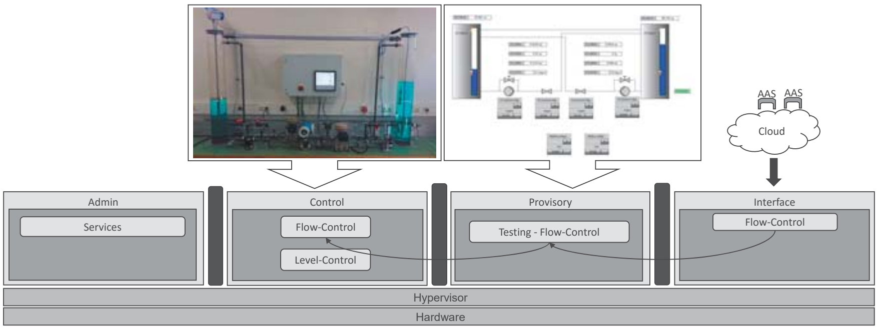

Fig. 7. Prototype Realization

contains a control partition that contains the control logic of the real plant, provisory partition that contains the simulation of the plant which serves as a platform for testing and validation purposes for the newly introduced control strategy. The interface partition communicates with the cloud in order to deploy the required control components.

In the next process steps, the volume of the liquid that flows from tank 1 to tank 2 should be controlled by means of a flow controller. In this case an operator or higher automation levels such as MES communicate to Admin partition that the level controller should be replaced by a flow controller. This triggers the deployment process of the control component to the external interface partitions which is allowed by the Admin partition. The flow controller should be tested and validated before being applied in control logic. This test is accomplished in provisory partition. After testing the flow controller it can be sent to the control partition in order to replace the existing strategy on the real plant control.

VII. Conclusion

Aspects like security, adaptability, scalability, etc., have imposed risks and difficulties integrating classic control systems into the newly introduced automation architecture in Industry 4.0. In this paper, PLC 4.0, a new infrastructure for control devices operation is introduced. This infrastructure comprises four meta classes: Admin, Control, Provisory and Interface. With the help of Hypervisor technology, we realize the meta classes thus isolating the different functionalities and carrying them out in different partitions. PLC 4.0 offers a seamless integration of PLC with concepts like DT, cloud computing, IoT, etc. We present a prototype implementation that uses a simple automation plant to demonstrate the concept.

References

[1] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, Marimuthu Palaniswami, Internet of Things (IoT): A vision, architectural elements, and future directions, Future Generation Computer Systems 29 (2013)

[2] The smart factory, https://www2.deloitte.com/insights/us/en/focus/industry-4-0/smart-factory-connected-manufacturing.html

[3] The smart factory, online: https://www2.deloitte.com/content/dam/ insights/us/articles/4051_The-smart-factory/DUP_The-smart-factory.pdf

[4] O. Givehchi, J. Imtiaz, H. Trsek and J. Jasperneite, ”Control-as-a-service from the cloud: A case study for using virtualized PLCs,” 2014 10th IEEE Workshop on Factory Communication Systems (WFCS 2014), Toulouse, 2014, pp. 1-4. doi: 10.1109/WFCS.2014.6837587

[5] G. J. popek and R. P. goldberg. formal requirements for virtualizable third generation architectures. Commun. ACM, 17(7):412-421, July 1974 T. Hegazy and M. Hefeeda, “Industrial Automation as a Cloud Service,” Parallel and Distributed Systems, IEEE Transactions on, vol. 26, no. 10, pp. 2750–2763, 2015.

[6] Thomas Goldschmidt, Stefan Hauck-Stattelmann, Somayeh Malakuti, Sten Grüner, ”Container-based architecture for flexible industrial control applications,” 2018 Journal of Systems Architecture, Volume 84, March 2018, Pages 28-36, https://www.sciencedirect.com/science/article/pii/S1383762117304988

[7] T. Hegazy and M. Hefeeda, “Industrial Automation as a Cloud Service,” Parallel and Distributed Systems, IEEE Transactions on, vol. 26, no. 10, pp. 2750–2763, 2015.

[8] Intel, Newsroom, https://newsroom.intel.com/news/new-breakthrough-wind-river-software-platform-control-systems-transforms-future-industrial-iot/, accessed: 21.09.2019

[9] F. Tao and M. Zhang, “Digital Twin Shop-Floor: A New Shop-Floor Paradigm Towards Smart Manufacturing,” in IEEE Access, vol. 5, pp. 20418-20427, 2017. doi: 10.1109/ACCESS.2017.2756069

[10] M. Grieves. (2014). Digital Twin: Manufacturing Excellence Through Virtual Factory Replication. [Online]. Available: http://www.apriso.com

[11] L. Monostori, “Cyber-physical production systems: Roots, expectations and R&D challenges,” Procedia CIRP, vol. 17, pp. 9–13, Jan. 2014.

[12] D. Mourtzis, E. Vlachou, N. Milas, and N. Xanthopoulos, “A cloud-based approach for maintenance of machine tools and equipment based on shopfloor monitoring,” Procedia CIRP, vol. 41, pp. 655–660, Jan. 2016.

[13] https://patents.google.com/patent/US9568908B2/en

[14] O. Agesen, A. Garthwaite, J. Sheldon, and P. Subrahmanyam. The evolution of an x86 virtual machine monitor. ACM SIGOPS Operating Systems Review, 44(4):318, 2010.

[15] Enterprise Container Platform for High-Velocity Innovation, Online:https://www.docker.com/

[16] M. Polke, U. Epple and M. Heim, Process Control Engineering, VCH Verlagsgesellschaft mbH, D-69451 Weinheim, 1994, ISBN: 3-527-28689-6.

[17] Christian Dufour, Zareh Soghomonian, Wei Li, Hardware-In-the-Loop Testing of Modern On-board Power Systems using Digital Twins, Opal-RT Technologies, 1751 Richardson, suite 2525, Montréal, Canada CACI International Inc, 300 M. Street, SE, Washington, DC 20003, USA

[18] R. Creasy, A. Jansen and A. JansenKristian Sandstrom¨ Virtualize for Architecture Sustainability in Industrial Automation, International Conference on Computational Science and Engineering, 2013

[19] Robert. P. Goldberg and S. Hauck-Stattelmann, “Survey of virtual machine research”, IEEE Computer Magazine, 1974, 7(6), pp. 34–45.

[20] Elisa Negri, Luca Fumagalli, Marco Macchi, A Review of the Roles of Digital Twin in CPS-based Production Systems, Procedia Manufacturing, 2017

[21] Dmitry Namiot, Manfred Sneps-Sneppe, On Micro-services Architecture, International Journal of Open Information Technologies, 2014