Unified Plug&Produce Architecture for Automatic Integration of Field Devices in Industrial Environments

Dipl.-Ing. Veit Hammerstingl
Institute for Machine Tools
and Industrial Management (*iwb*)
Technical University of Munich
Boltzmannstr. 15
D-85748 Garching
Email: Veit.Hammerstingl@iwb.tum.de

Prof. Dr.-Ing. Gunther Reinhart
Institute for Machine Tools
and Industrial Management (*iwb*)
Technical University of Munich
Boltzmannstr. 15
D-85748 Garching
Email: Gunther.Reinhart@iwb.tum.de

***Abstract*—By introducing approaches like the "Smart Factory" and the "Internet of Things", companies try to ready their manufacturing systems for the needs of the future. The aim is to increase the changeability of these systems through intelligent and self-configuring devices. Prerequisite is an appropriate communication capability down to the lowest sensor level, which is, however, difficult to achieve in current industrial environments. This article describes the origin of this complexity and the requirements future factories will have to meet. Based on the analysis of current industrial environments, a system architecture is developed, allowing the automatic integration of field devices from heterogenous communication networks. The concept is compatible with existing devices and presents a unified information model for the standardized data exchange with other information systems.**

Keywords: Plug and Produce, Smart Factory, control systems, CPS, legacy systems

I. Introduction

In order to manufacture economically in high-wage countries, a high degree of automation is essential. However, companies are facing an increasingly turbulent market environment, where systems must be adjusted more frequently [1]. This need for reconfiguration is becoming a problem due to the resulting high costs during the commissioning phase [2].

The main reason for these high costs are the long start-up times and the necessity of using experts. The market for automation equipment is very fragmented and many niche markets exist. There is a lack of cross-technology standards on different software levels, which results in a high degree of heterogeneity regarding description standards, programming languages and vendor-specific tools. As the commissioning of field devices in factories still requires the different installation steps to be done manually, system integrators have to deal with this heterogeneity. This also encompasses the knowledge of various communication protocols, commands closely tied to the hardware and the parallel usage of various vendor specific software tools [3]. As a consequence, companies try to avoid any adaption of their production machinery and build automated systems for the purpose of mass production.

This is in contrast to the megatrends, where product life cycles are becoming more indefinite [4]. Since modern communication channels (Social Media, microblogging) help trends spread much faster and global competition puts companies under a lot of pressure to come up with ever new inventions, developing a new manufacturing system for a product often can no longer be regarded as economical. As a response, a design referred to as "reconfigurable manufacturing systems" (RMS) has emerged. These systems are characterized by their high degree of changeability, which allows them to be quickly adjusted to unforeseen requirements even after already having been put into operation [5]. The changeability can pertain to different aspects of the system such as the adaption to different variants, process parameters or other levels of automation [4]. Concerning field device integration, this concept has also come to be known as Plug&Produce (as well as plug&work or plug&play) [6], which implies an uncomplicated and fast integration of the device. Current studies in this regard mainly concentrate on creating suitable interfaces by means of software algorithms rather than by electrical or mechanical means, which are more complicated to automate.

This article describes a concept showing how a Plug&Produce architecture can be used for an automated integration of field devices. First, the installation steps currently needed for control and data transfer are introduced. Related work for automating these steps is shown, and general requirements for an automatic integration are derived. Additionally, demands of future smart factories on Plug&Produce device integration are given. Following the premise of developing a concept which is compatible with current plant designs, a technological analysis of a standard automation architecture is done. The result of these considerations is a concept that allows an automatic integration of heterogeneous field devices into superior logic controllers and publishes the available information in a unified manner.

II. State of the art of field device integration

The use of field devices is subject to the typical stages in the plant life cycle: planning, construction, commissioning, operation and decommissioning [7]. Because of the developments mentioned above, manufacturing systems have to be adjusted more frequently. Thus, these phases are repeated in a cyclic manner, with increasing speed.

However, only the fourth phase is value added; phases one to three are preparatory. One of the highest aims is to drastically reduce the cycle times of these preparatory phases. The planning phase is mainly concerned with organizational / administrative tasks and information technology (IT) tools with a high level of abstraction. The construction phase mainly concentrates on mechanical and electrical tasks. This article focuses on the commissioning phase, which is primarily software driven.

The following steps have to be carried out by a system integrator each time a field device is put into operation.

- S1 **Connection** First, a physical communication link has to be established. For simple sensors this can be an electrical wiring of 24 volts or a modular connector such as the RJ45 socket for Ethernet-based buses.
- S2 **Device recognition** To establish communication, the device must then be made known to the communication master (usually a programmable logic controller, PLC). When handling traditional fieldbuses, a bus address is manually set by the user in the corresponding engineering tool of the master and is saved on the device. Occasionally modern Ethernet-based fieldbuses (Real-time Ethernet, RTE) can search for connected devices automatically. In addition, many components still need the setting to be adjusted by a direct input, e.g., via buttons or coded rotary switches.
- S3 **Configuration of the communication protocol** In this step the signal coding between the communication master and the field device is determined. Data types, bit length and time behavior has to be entered manually in the engineering tool of the communication master. To facilitate this process, manufacturers also provide device description files that already have this information stored. These are hardware- and protocol-specific, so the user has to select the correct communication file for the currently used fieldbus (e.g., GSD for Profibus, EDS for EtherNet/IP, etc.), select the correct version for the corresponding model and load it into the master tool.
- S4 **Signal mapping** In this phase, the programmer maps the process data at the in-/outputs of the controller (this also includes data that is transmitted over fieldbuses) onto the variables in the control program. Once a variable changes in the PLC, a change will also take place in the field device via the process variables configured in S3 and vice versa.
- S5 **Parametrization** In general, each device must be configured depending on the current application, for example the maximum stroke. For this purpose, either proprietary software tools or vendor-independent standards such as FDT [8] are used. The connection needed for the parametrization can be achieved by plugging the PC directly to the device (e.g., USB or RS-232) or remotely via the network.
- S6 **Definition of the application** In this step, the process sequence of the manufacturing system is programmed using logic operations and the I/O signals. To that end, the user utilizes the engineering tool of the PLC where he can encapsulate signal operations generating more abstract function blocks for easier handling and reuse.
- S7 **Upload of the software** After the behavior of the system has been described for all possible states, the software can be uploaded by the engineering tool to the PLC hardware. The access to the hardware is usually encapsulated by the manufacturer for which reason the engineering tool must generally be used.

The sequence shown above represents the logical procedure. Some activities, such as the programming of the system behavior or the parameterization of the devices, are partly carried out offline prior to commissioning. Yet very few offline created programs can be transferred directly onto the PLC without adjustments.

The described steps show a high degree of manual activities that need to be conducted by an expert. For an automatic integration of field devices, consideration has to be given as to how the steps described above can be automated to a higher degree.

III. Related Work

There are several research contributions that investigate the field of Plug&Produce integration in manufacturing systems.

[9] presents a method for an ad-hoc field device integration into logic controllers. Devices offer semantic information about their functionalities via device profiles for web services (DPWS, [10]). Therefore the field device is enhanced with an additional microcontroller board that offers the services. The configuration software searches for these profiles in a standard Ethernet network and matches them with the modeled production steps. [11] focuses on the automatic configuration of robotic controllers in real-time Ethernet networks. There, intelligent sensors and actuators each have an HTTP-server, which offers the functionalities and protocol settings of the device. A configuration software collects and integrates this information in the communication master as well as in the robotic application. A second approach shows the automatic configuration of an Ethernet Powerlink network, without the need of additional servers. [12] presents an approach for a service-oriented architecture (SOA) in industrial automation systems. Several devices are combined with a logic controller to form a “mechatronic module”. The modules communicate via web services over a standard Ethernet network. Inside the modules, each field device has its own OPC UA server which is discovered by a search client of the modules controller. After signal information was downloaded from the individual device servers, an RTE communication between

the controller and field devices can be established. In [13], manufacturing stations contain different kinds of models describing the internal structure and the supported processing steps. The station models are dynamically combined to a factory model and matched against a product plan model to schedule different manufacturing tasks. Each station contains a PLC, which communicates via a middleware with other systems. The developed middleware handles the automatic recognition and configuration of the communication partners. It can deal with heterogeneous communication technologies, as long as they are implemented (currently: Ethernet and MPI). Statements on the real-time capability are not apparent. Other projects, like [14]–[16], focus on the possibilities of dynamic reconfiguration in a more abstract way, while not specifying how a technological implementation could be accomplished.

The presented research projects show good approaches for lowering the needed configuration and programming efforts. However, the focus of most projects is usually above the PLC level; the integration of field devices in PLC controllers is rarely considered. Furthermore, the aspect of a real-time capable execution and communication is often disregarded as is the heterogeneity of current manufacturing architectures. In addition, used field devices are often modified to provide certain interfaces which are currently not used on a broader scale. There is a need to develop a concept which takes these boundary conditions into consideration.

IV. Requirements for the Next Generation Factory

Future factories will be characterized by a high changeability of their production processes. This can be achieved by either an adjustment of the real (physical) manufacturing system or the virtual system (software). On a physical level, this means the automation of the commissioning steps S1 – S7.

This results in the following general requirements for Plug&Produce:

- R1 Automated setup of a communication link and announcement at the communication master (S1, S2).
- R2 Automated configuration of communication data for participants to exchange process data (protocol) (S3).
- R3 Software interface to remotely set up field devices for their specific task (parameterization) (S5).

The virtual changeability will be achieved by novel planing algorithms, which can – based on data of smart machines – dynamically change, alter and reschedule production processes. This is made possible by an ad-hoc connectivity of all relevant production resources and the real-time availability of all device information [17].

Plug&Produce concepts will form the basis to deliver this ad-hoc connectivity. If, for example, an additional robot is integrated into the manufacturing system, the future factory can immediately include this new resource in its production chain and adjust other processes accordingly. For this purpose, the algorithms require extensive information about the logical and physical state of the device (e.g., geometry, status, position). This virtual representation (VR) must be digitally existent and accessible for each device and extends far beyond the current description of the communication aspects.

To ensure that the high degree of heterogeneity at the field level does not have to be repeated on the IT level, it is necessary that the access to the field devices is independent of communication channels and hardware used. It appears to be logical to deduce a unified model of the field level, where devices offer their services in a standardized manner. IT systems can use these services to retrieve and control different plant conditions. In [18], this is mentioned as the “Diabolo Model”. This also meets the requirements of RMS, two of which being modularity and compatibility with standardized software interfaces [19].

Due to the often necessary parallel control of processes and for various safety reasons, the execution of the process steps as well as the communication to the field devices must be deterministic.

This leads to additional demands by the future factory for Plug&Produce:

- R4 Provision of an extended virtual description (virtual representation) of field devices. Other software programs can utilize this for the generation of planning and control data.
- R5 Abstract model of the manufacturing system. The model must present standardized services from field devices to enable a protocol- and hardware-neutral exchange of information with these.
- R6 Interface to the control system (e.g., PLC) in order to import new or modified production processes. Furthermore, the interface must have access to the process data mapping (S4, S6, S7).
- R7 The process control and communication with the field devices must be capable of real-time behavior.

All information-based Plug&Produce concepts should meet these general requirements. However, individual concepts may have additional demands, which need to be met.

V. Technological Analysis of a Standard Automation Architecture

For the development of a Plug&Produce concept that is suitable for the current situation in plants, today’s technical circumstances have to be analyzed more thoroughly. To achieve this, a representative system architecture is developed. Components of such a system structure are the communication architecture, communication protocols used, the communication partners involved and the definition of the data exchanged.

In order to determine the most relevant basic communication architecture, the survey from [20] is considered. Of nearly 400 companies surveyed, 74 percent stated that they prefer a centralized architecture. This has increased by 3.4 percent over the last 4 years. In addition, the vast majority of companies see PLCs that are based on IEC 61131-3 languages as the most commonly used, even for the future. Consequently, the future concept will have to handle a centralized architecture based on traditional PLCs.

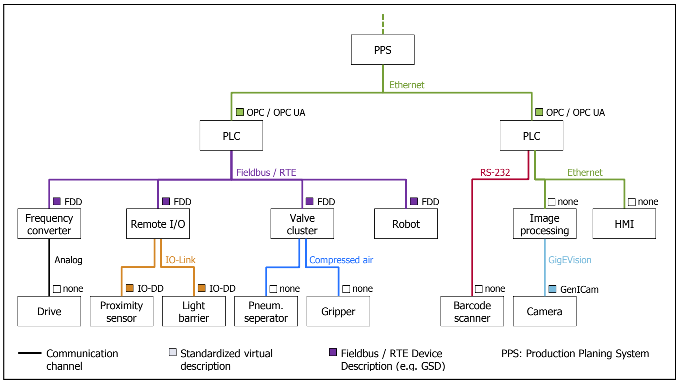

Fig. 1. Representative automation structure

For an analysis of the remaining three elements, an own survey was carried out with 12 companies questioned. The Focus of the study was the field of assembly technology. The companies were asked which device classes they use in assembly lines, over which protocols these communicate and what virtual descriptions are applied. Subsequently, the answers given were transfered into a representative system structure that reflects all individual characteristics (see Fig. 1).

The described systems can be divided into four classes. Primitive sensors and actuators represent the first class. These are directly wired (e.g., 24 volts or compressed air), the transmitted information being of purely physical nature (binary or analog). They do not provide digital descriptions; thus the host does not know which device is connected to it. A parametrization must be performed directly on the field device, e.g., via buttons or coded rotary switches.

The second class is represented by basic devices. These too are wired directly, with a minimum protocol running, which makes the identification of the connected device possible. In addition to binary and numeric values, also data of the type “word” can be transferred. This allows an external parameter setting. Information on the parametrization and process data segmentation are provided in structured files. A common representative of this category is IO-Link, along with its device description IO-DD. An exceptional position holds RS-232, which is usually used for older devices. A machine readable description of the specific protocol in use would be technologically possible, but is not applied in practice. The protocol settings have to be determined by the user or implemented by proprietary tools of the equipment manufacturer.

The next categories include devices that communicate over a bus protocol. Standard Ethernet [21] is one of the best-known representatives worldwide. Because it lacks real-time properties, it is usually used for process visualization and data exchange with higher IT levels. The main representative regarding field devices are image processing systems. Here, image data is sent from the camera via the GigE-Vision protocol, which is based on UDP, to the processing system. Virtual descriptions exist in the field of image processing systems by the GenICam standard, which describes the characteristics of camera hardware. However, abilities of the processing software are not included in this norm. In the field of deterministic bus communication, traditional or Ethernet-based fieldbuses (RTE) exist. Being mostly incompatible with each other, over 25 different protocols are currently used in the field of factory automation [22]. Due to long investment cycles of plants and the given history in this area, it is not expected that a consolidation will occur in the near future. As for a virtual description, mostly communication files are used, which can be imported into the bus muster in order to configure the process data segmentation (e.g., GSD). A parametrization is generally done through proprietary vendor tools (RTEs) or the FDT standard (traditional fieldbuses), where each vendor delivers an executable driver which can be loaded as a plug-in into a frame application [8].

In the recent past, attention shifted towards devices which have an integrated information server. In contrast to the other device categories, the server offers a current and complete virtual representation of the device in the sense of a smart factory (see Section IV). The contents of this virtual description are currently the subject of research. Other systems can access these devices over standardized interfaces. By enriching the published data with semantic information, external systems can issue queries about the content without having knowledge about the specification of the device. Since this correlates with the development of the “Internet of Things”, it can be assumed that this type of equipment will be the next generation of field devices (NGD). The implementation of the server can be done for example by technologies such as DPWS or OPC UA. Unlike web services, OPC UA is currently integrated in a large number of PLCs on the market. The IEC standardization commission recommends OPC UA as a standard for the implementation of a smart factory [23]. For this reason, OPC UA is used as server standard for NGDs. However, OPC UA does not allow real-time transmission, which is why a real-time communication channel must still exist. PLCs can be partially viewed as NGDs, since they already provide an OPC or OPC UA server along with a real-time communication channel. The integration of control programs into PLCs is mostly done by the engineering tool. For this purpose, the standard PLCopenXML [24] allows a multi-vendor exchange of IEC 61131-3 software applications. However, this virtual description focuses purely on the logical part; the hardware configuration and the mapping are not included. Table I shows an overview of the mentioned device classes.

As a result of the technical analysis, the following special requirements can be derived for a Plug&Produce concept that is compatible with existing architectures:

C1 The concept has to deal with devices that cannot identify themselves. Mechanisms have to be found to deliver these IDs externally.

C2 Since most device classes do not provide a complete virtual representation and this also cannot be retrofitted, the concept has to accommodate the missing parts of the VR.

C3 For the configuration of the communication master, the concept must be able to import established device

TABLE I
CLASSIFICATION OF DEVICES IN AUTOMATION SYSTEMS

Device class	Communication architecture	Communication predictability	Autom. ID retrieval	Interface of virtual device description	Transfered data	Connection	Example
Primitive	Direct wiring	Deterministic	No	None	Bool, numeric	Electrical wire, compressed air	Light sensor, gripper
Basic		Deterministic	Yes	File (e.g., IO-DD)	Bool, numeric, word	IO-Link, (RS-232)	Distance sensor
Standard	Bus	Non deterministic	No	None	Bool, numeric, word	Ethernet	HMI
Field		Deterministic	Yes	File (e.g., GSD), Driver (e.g., FDT)	Bool, numeric, word	Fieldbus, RTE	Robot, frequency converter
NGD		Deterministic and non deterministic	Yes	Online Server for accessing capabilites	Bool, numeric, word, complex	RTE (process), Ethernet (config.)	PLC

description standards. This allows a downward compatibility with existing means of communication.

C4 Since the information model provides the foundation for many external IT systems, it must deliver access through widely accepted standards.

C5 The concept must handle different heterogeneous characteristics that are present in form of communication, device intelligences, determinism classes and means of description.

In order for the found requirements to be met, different solutions have to be found. The aim must be that, besides the physical connection and installation of the device, the user effort tends to zero and the necessary settings are carried out automatically by a software system.

VI. CONCEPT

The current process state is represented by digital information on the field devices. These must be loaded from the device and published in an abstract way in the information model.

For this purpose, the device may already have a standardized server which publishes its services (like NGDs). Its internal logic converts the hardware-specific data into standardized information on the device server. The business logic (BL) of the Plug&Produce configuration tool would therefore only have to link the services to the information model (see Fig. 2, left side).

However, the overwhelming majority of field devices in current factories do not have this technological ability, for which reason the business logic has to perform the task of abstraction. Because the information access as well as the syntactical and semantical conversion of the information is specific for each field device model, these hardware-specific calls are transfered into separate drivers. This allows the BL to access information in a standardized manner. Due to the necessary conversion, the business logic acts as an intermediary between the device and IM and must therefore ensure the synchronization between the two levels. Once the information model is populated, it gives a current image of the abilities and properties of the manufacturing system. The content and structure of this overall virtual representation is currently the subject of research and lies outside of the scope of this paper.

Higher-level systems, of technical or human nature, can now use the model. For example, in the planning stage a user can define and model the necessary manufacturing steps and its requirements of a certain production sequence. This can be done independently of the later used hardware. These abstract manufacturing steps can subsequently be compared with the currently available device services from the information model. Appropriate services respectively hardware can be selected by planning algorithms or by the user, who, for example, is offered a selection of suitable devices for each production step.

Afterwards, the generated sequence of service calls can be uploaded to the process control (e.g., PLC). For that purpose the abstract service calls have to be converted into specific commands of the controller. To do so, the corresponding engineering tool of the controller is used. This has the advantage that the user can make adjustments in his usual tool environment and even use existing code parts. Moreover, the complex checking and compilation process is still left entirely to the tool. If an IEC 61131-3 based PLC is used, an example would be a conversion to "Program Organization Units" (POU). Like the services, the POUs encapsulate the specific technical control of the device, giving the user the possibility to adjust the application on a purely functional level. After the program adjustments are completed, the code must be transfered from the engineering tool to the controller (Fig. 2, right). Variables inside the POUs, which address field device data, have to be mapped to the process data variables (dynamic mapping). This can be done, for example, by a simple name-based matching. Since there are no common industry standards for accessing PLCs or their engineering tools, a manufacturer-specific driver (Host Driver) is needed that converts the general BL-calls into corresponding manufacturer commands.

In the last step, the controller is started, which then executes the imported process sequence and communicates with the field devices, both in real-time.

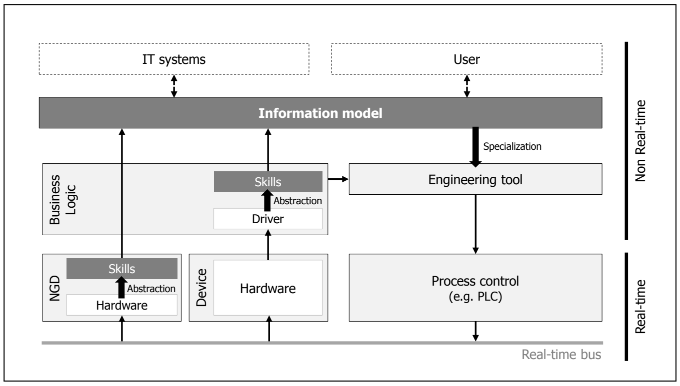

Fig. 2. Basic proceeding of the concept

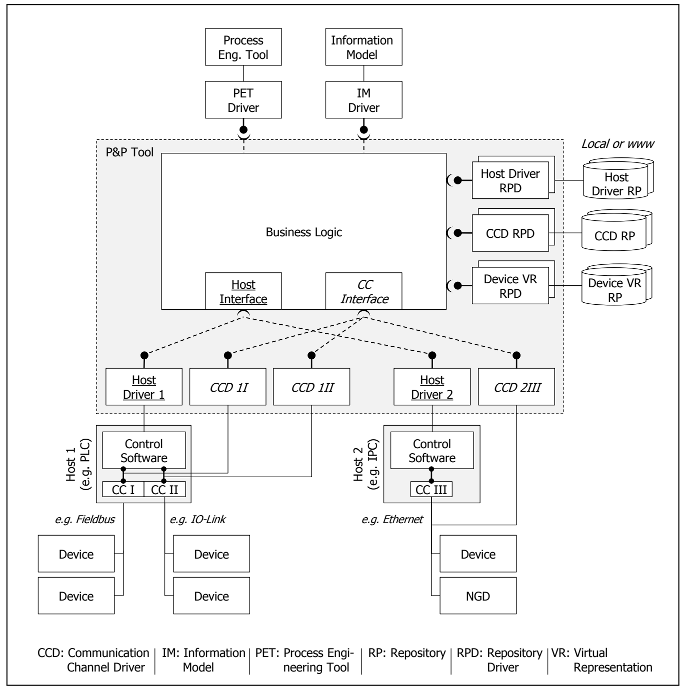

Fig. 3. Architecture of the Plug&Produce concept

Before a device driver can access information on the field devices, they first have to be detected on the communication channel. This requires two steps for each communication channel to be carried out (NGDs are excluded):

1) *Discover devices:* Connected and active devices have to be detected by mechanisms such as broadcasts.
2) *Retrieve basic identification:* The goal is to establish a minimum connection to the device in order to retrieve basic information such as serial or model number.

These steps correlate roughly with step 2 and step 3 from the automatic configuration process according to [11].

This would mean a physical access to each communication channel as well as an implementation of each protocol used. It is noticeable that the PLC already has access on the physical and the software level for its future control tasks. This allows the PLC to be utilized for accessing the communication channels. Since control systems are modular and therefore can offer different ports and protocols, it seems reasonable, considering the design principle of "separation of concerns", to provide each communication channel with its own driver, which in turn provides access to the PLC communication channel. The Host Driver together with the Communication Channel Drivers (CCD) forms an overall hardware access package for the specific host (see Fig. 3). In addition to modularity, the advantage of this is that the business logic of the Plug&Produce software can use a definite interface and can thus perform identical calls regardless of the communication channel. This leads to a generic and well scalable solution which makes it possible to search for available hosts; in turn for each host available communication channels can be found, and for each channel available devices. For every detected device, a basic communication is established, via which a unique identifier of the device is requested (e.g., combination of serial number and manufacturer ID). Fieldbuses and RTEs often offer this possibility inherently. If not (e.g., primitive devices), the user must select the identifier manually in the HMI of the configuration tool (C1).

With this ID, the business logic can load the associated virtual representation (VR) of the device from a repository (RP) and integrate its information into the information model (see Fig. 4). Elements of the VR can be, for example, the process data descriptions such as GSD (Profibus) or IO-DD (IO-Link). Further elements are offered services, time behavior and current state of the device. The data which has to be additionally implemented depends on the already existing means of description for the device (C2). Definition and scope of the virtual representation are currently being researched and will be addressed in more detail in a subsequent paper. With the information from the VR, the business logic accesses the controller by means of the host driver and configures the I/O data of the respective device. This can usually be done by automated import of the standardized process data description files (C3). Subsequently, a notification channel has to be created on the PLC for each dynamic process value of a device. If the OPC (UA) server of the PLC is used, this can be done, for example, by means of subscriptions or, if an API is available, by creating events. Changed values of the field device are recognized by the PLC and passed on to the BL via the relevant mechanism, which writes the changes in the information model and thus keeps it synchronized.

The Communication Channel Driver does not always have to access the PLC hardware directly. For well-known protocols such as Ethernet, the access can be implemented by the user himself (see Host 2 in Fig. 3). This would yield the advantage of independence between the controller and the used communication driver. Generally effort of implementing the communication protocol and efforts of accessing the PLC hardware have to be weighed against each other. For industrial PCs, which have very uniform operating systems and interfaces (USB, Ethernet), the second variant seems more reasonable; for traditional PLC systems, the first option makes

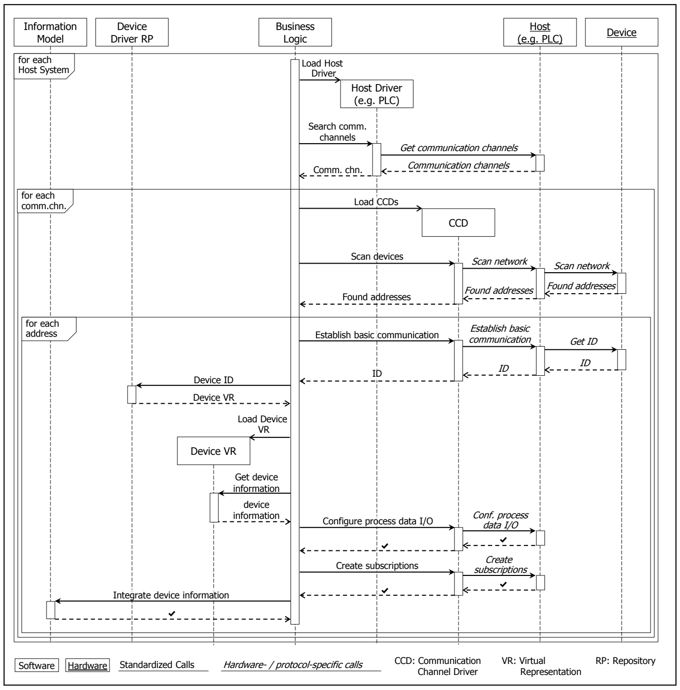

Fig. 4. Sequence of configuration

more sense.

Following the recommendation of the IEC [23], the standard OPC UA is used for the information model (C4). The data which is exchanged between the field devices and information model is completely decoupled by the use of the driver concept and its generic interfaces (C5). In general, all elements of the business logic that access external systems are abstracted by using interfaces in order to achieve a loose coupling and easy changeability in the sense of Plug&Produce.

Based on different standings, two ways by which external systems may interact with the presented architecture are conceivable: On the one hand, the architecture could extend to become a single global information model of the factory, in which each of the systems are integrated. On the other hand, it could be possible for every subsystem to have its own information model. Based on the definition of [25], the presented architecture could be seen as one cyber-physical system (CPS), which internally acts on different information signals, while it presents its capabilities to other CPSs. Each CPS would have its own sub-architecture, so multiple information models would exist at the same time. Which of these approaches can be considered best will have to be determined through further research.

VII. Plug&Produce validation

To demonstrate the feasibility of this concept, an exemplary architecture has been developed, which corresponds to the left side of the representative automation architecture in Fig. 1. It consists of a standard PLC CX2030 from Beckhoff. Connected to this via the real-time Ethernet protocol EtherCAT are an IO-Link master EP6224 and a frequency converter VLT-FC 302 from Danfoss. At the IO-Link master, a light sensor MLV41-8-H from Pepperl&Fuchs and an inductive distance sensor BAW M18MI from Balluff are attached (see Fig. 5).

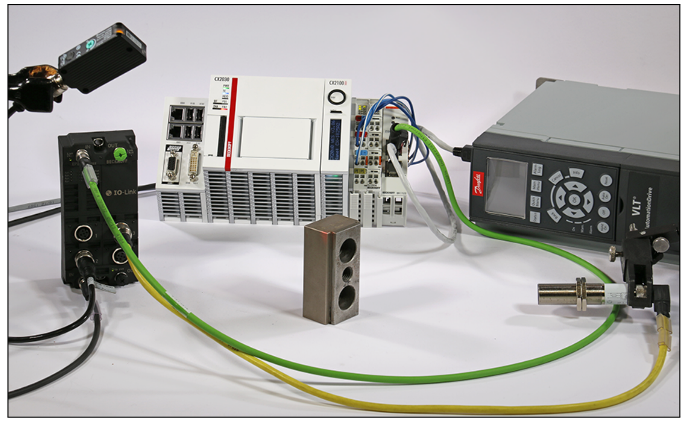

Fig. 5. Hardware installation of the validation demonstrator

All of the equipment meets the requirement R1. The user can initiate a search for devices via the HMI of the configuration tool. First, the business logic searches in the connected Ethernet network for host systems (Fig. 4, Step 1). The Beckhoff PLC is identified and the corresponding Host Driver is loaded. In the next step, the existing communication channels of the PLC are requested. For this purpose Beckhoff delivers two APIs (Automation Runtime and ADS) to obtain access to the engineering software and the PLC. Other PLC vendors also provide external access, e.g., Bosch with its Open Core Interface or CoDeSys with its PLCHandler, so that R6 is considered as fulfilled. The Communication Channel Drivers can be used to search for device IDs in a generic way. For example, the business logic sends identical commands to the EtherCAT CCD and the IO-Link CCD.

By means of the returned IDs (e.g., MLV41-8-H for the light sensor), the corresponding virtual representation of the device is loaded from a repository (R4). In this case, this includes the files for the process data configuration (IO-DD for the IO-Link sensors and XML for the EtherCAT devices) as well as the published services (e.g., "Measure distance" etc.). The communication files are imported automatically into the engineering tool TwinCAT V3, where also the process data mapping is configured (R2). The device services are converted into IEC 61131-3 compliant POUs and integrated into the engineering tool. An abstract process sequence is not modeled for this validation. Adjustable parameters, like the measurable distance of the light sensor, are extracted from the IO-DDs and displayed on the HMI. The user can make changes to these, which are forwarded to the IO-Link CCD, from where they are transmitted via the API to the PLC and finally to the devices (R3).

In the next step, subscriptions are created via ADS on the generated POUs, so that the business logic receives a notification every time a value has changed on the PLC, which is then displayed on the HMI. A connection to an abstract information model implemented in OPC UA is currently being researched (R5). Nevertheless, the synchronous display on the HMI shows that a link between the information model and the field level is possible.

Completing the configuration process, the BL starts the PLC via the Host Driver, resulting in a real-time capable execution of the integrated POUs (R7). Although the information model reflects the current state of the field level, it cannot be used for a live control of the field devices due to the non-real-time capability of the configuration software. If the production sequence is to be changed, the altered control steps must be uploaded to the PLC again.

VIII. Conclusions and Outlook

The demands regarding automated integration of field devices in terms of Plug&Produce have still not been met. Based on the steps needed to integrate field devices into logic controllers, requirements for automated implementation were derived. These were supplemented by the requirements of future factories concerning the fields of modularity (abstraction) and virtual representation. In order to make a concept compatible with existing industrial architectures and devices, a representative automation architecture was developed. Based on this, an analysis was performed, showing the large technological heterogeneity in automation systems at different levels. The developed concept automates the obligatory configuration steps and encapsulates the heterogeneity through a multi-stage driver concept, with the interfaces of existing control hardware being used to keep implementation efforts low.

In addition to automatic integration, the concept shows how a unified virtual representation of the manufacturing system can be composed of heterogeneous technological sources. This can be used by higher-level IT systems as a base model to retrieve information about the current state of the field level as well as a means to plan production sequences according to available services. Thus, the concept makes a contribution toward bringing existing plants closer to a smart factory.

Further work must be carried out in the area of virtual representation of devices. It has to be analyzed which information superordinate algorithms request dynamically from field devices and how a meta-model description can be organized. Another aspect is the necessary parameterization of field devices. For maximum changeability of the system, a higher degree of automation has to be realized. The aim is to develop methods through which field devices can autonomously adapt to new process requirements and environmental conditions, thus bringing Plug&Produce manufacturing systems closer to reality.

Acknowledgment

The authors would like to thank the Bavarian Ministry of Economic Affairs and Media, Energy and Technology for funding this work, which is part of the "Automatic Configuration in Microsystems Technology (AKOMI)" project.

References

[1] E. Abele and G. Reinhart, *Zukunft der Produktion: Herausforderungen, Forschungsfelder, Chancen*. München: Hanser, 2011.

[2] G. Reinhart and S. Krug, "Current state model for easy reconfiguration of robot systems and offline-programming-environments," in *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*.

[3] "'One Ethernet' concept," *industrial ethernet book*, no. 07, p. 4, 2013.

[4] H.-P. Wiendahl, H. A. ElMaraghy, P. Nyhuis, M. F. Zäh, H.-H. Wiendahl, N. Duffie, and M. Brieke, "Changeable manufacturing - classification, design and operation," *CIRP Annals - Manufacturing Technology*, vol. 56, no. 2, pp. 783–809, 2007.

[5] M. G. Mehrabi, A. G. Ulsoy, and Y. Koren, "Reconfigurable manufacturing systems: Key to future manufacturing," *Journal of Intelligent Manufacturing*, vol. 11, no. 4, pp. 403–419, 2000.

[6] M. Naumann, K. Wegener, R. D. Schraft, and L. Lachello, "Robot cell integration by means of application-p'n'p," in *ISR 2006 - ROBOTIK 2006: Proceedings of the Joint Conference on Robotics*.

[7] M. Mertens and U. Epple, "Plant asset management functions driven by property models," in *Factory Automation (ETFA 2009)*, pp. 1–8.

[8] "Field device tool (FDT) interface specification," IEC 62453-1, 06/2009.

[9] S. Hodek and J. Schlick, "Ad hoc field device integration using device profiles, concepts for automated configuration and web service technologies: Plug&play field device integration concepts for industrial production processes," in *2012 IEEE 9th International Multi-Conference on Systems, Signals and Devices (SSD)*.

[10] Devices profile for web services (DPWS): Version 1.1. (2009) [Online]. Available: http://docs.oasis-open.org/ws-dd/dpws/wsdd-dpws-1.1-spec.html

[11] G. Reinhart, S. Krug, S. Hüttner, Z. Mari, F. Riedelbauch, and M. Schlogel, "Automatic configuration (plug & produce) of industrial ethernet networks," in *2010 9th IEEE/IAS International Conference on Industry Applications - INDUSCON 2010*.

[12] L. Dürkop, H. Trsek, J. Otto, and J. Jasperneite, "A field level architecture for reconfigurable real-time automation systems," in *2014 10th IEEE Workshop on Factory Communication Systems (WFCS)*, pp. 1–10.

[13] N. Keddis, G. Kainz, C. Buckl, and A. Knoll, "Towards adaptable manufacturing systems," in *2013 IEEE International Conference on Industrial Technology (ICIT 2013)*, pp. 1410–1415.

[14] J. Michniewicz and G. Reinhart, "Cyber-physical robotics – automated analysis, programming and configuration of robot cells based on cyber-physical-systems," *Procedia Technology*, vol. 15, pp. 567–576, 2014.

[15] J. Malec, A. Nilsson, K. Nilsson, and S. Nowaczyk, "Knowledge-based reconfiguration of automation systems," in *2007 IEEE International Conference on Automation Science and Engineering*, IEEE, Ed., pp. 170–175.

[16] M. Bengel, "Modelling objects for skill-based reconfigurable machines," in *Innovative production machines and systems*, E. E. Eldukhri, D. T. Pham, and A. J. Soroka, Eds. Whittles Publ, 2008, pp. 238–243.

[17] D. Zuehlke, "Smartfactory—towards a factory-of-things," *Annual Reviews in Control*, vol. 34, no. 1, pp. 129–138, 2010.

[18] B. Vogel-Heuser, G. Kegel, K. Bender, and K. Wucherer, "Global information architecture for industrial automation," *atp - Automatisierungstechnische Praxis*, vol. 51, no. 01-02, pp. 108–115, 2009.

[19] H. A. ElMaraghy, "Flexible and reconfigurable manufacturing systems paradigms," *International Journal of Flexible Manufacturing Systems*, vol. 17, no. 4, pp. 261–276, 2005.

[20] M. Rothhöft, "Marktstudie SPS-Systeme: Ergebnisse einer bundesweiten Befragung von Unternehmen aus den Bereichen Maschinenbau und Ingenieurbüros im Januar/Februar 2014," Hemer, 2014. Available: http://www.marktstudien.org/marktstudien/marktstudie_sps.html

[21] "IEEE standard for Ethernet," IEEE 802.3, 2012.

[22] F. Klasen, V. Oestreich, and M. Volz, Eds., *Industrial communication with fieldbus and Ethernet*. Berlin and Offenbach: VDE-Verl, 2011.

[23] J. Kroll. (14.05.2014) IEC-Meeting: Normung für Industrie 4.0. [Online]. Available: http://www.computer-automation.de/feldebene/vernetzung/artikel/108790/1/

[24] "XML formats for IEC 61131-3: Technical paper - PLCopen Technical Committee 6: Version 2.01 - Official release," 2009.

[25] J. Jatzkowski and B. Kleinjohann, "Towards self-reconfiguration of real-time communication within cyber-physical systems," *Procedia Technology*, vol. 15, pp. 54–61, 2014.