A PLC as an Industry 4.0 component

Reinhard Langmann, Leandro F. Rojas-Peña
Competence Center Automation Düsseldorf (CCAD)
Düsseldorf University of Applied Sciences
Düsseldorf, Germany

Abstract: *This paper presents a concept for the use of a PLC as an Industry 4.0 (I40) component as defined by the Industry 4.0 architecture RAMI 4.0. As a result, PLC controllers can be integrated seamlessly in Industry 4.0 production environments using the service paradigm. The transmission time for the process data from the PLC to the IP network was determined for a prototype implementation. With the PLC as an I40 component, process data can be transferred directly from the controller to an IP network without additional delays. This forms, amongst other things, the basis for outsourcing parts of the control program to a cloud.*

Keywords: *Industry 4.0 component, PLC controller, web-based control system, device gateway, automation system.*

I. INTRODUCTION

Industrial controls and, in particular, PLC controllers currently form an important technological basis for the automation of industrial processes. Even in the age of Industry 4.0 (I40) and Industrial Internet, it can be assumed that these controllers will continue to be required to a considerable extent for the production of tomorrow. However, the controllers must fulfil a range of additional requirements resulting from the new production conditions.

When applying Industry 4.0 principles [1], high-quality networked production systems result based on Cyber Physical Systems (CPS), also referred to as Cyber Physical Production Systems (CPPS). A series of I40 requirements are placed on the future controllers used in these systems. These include:

- Autonomy, reconfigurability and agility (Plug&Work);
- Overcoming the strict information encapsulation of controllers;
- Introduction of the service paradigm in the production automation (production services);
- Networking in local and global networks;
- Interoperability between heterogeneous control systems;
- Dependencies are to be changeable dynamically to the runtime;
- Use of models for the development of "higher-quality" control approaches;
- Orchestration of heterogeneous controllers.

Current PLC controllers cannot yet fulfil the majority of these requirements or can only do so on a rudimentary basis or with extremely high expense.

One of the basic requirements of future and I40-able PLC controllers involves efficient networking in an, at least partially, global network. Here the IP network (IP – Internet Protocol) functions as a global network in the version as Intranet or Internet with all associated standardized Information and Communication (IC) technologies. Only in this way can the required integration become part of a future I40 production landscape.

This paper will concentrate on the basic requirement of the global networking and describe the concept and a prototype implementation for a PLC controller as I40 component.

II. State-of-the-art

Resulting from the historical development of PLC controllers, these have been developed as proprietary device systems that are operated locally under real-time conditions. If a networking of these controllers is necessary from a user viewpoint, proprietary TCP/IP protocols or those standardized in the automation sector (Modbus TCP, Profinet etc.) are used for this. The standard technologies widespread from the Internet and Web have so far played hardly any role for PLC controllers.

For a number of years, however, a transformation has been under way, with PLC manufacturers increasingly integrating IC technologies from the web world in their systems, such as web server and HTML pages for diagnosis and configuration, in order to adapt the controllers incrementally to the new requirements.

Three different approaches to make PLC controllers I40 compatible can essentially be revealed from the state-of the-art. These include the introduction of basic web technologies, the global networking of process data and the introduction of service principles.

A. Introduction of basic web technologies

Most of the newer PLC controllers already contain a web server as well as special HTML pages on the device, these enabling a browser-based configuration and diagnosis of the controller. Process data or program variables form the control program and can also be read, and sometimes also written, with restrictions. Access via a web browser is via the HTTP protocol and is hence query-based and relatively slow. Examples of this can be found in [2]. The solutions are proprietary and adapted to the relevant controller. Open and consistent web interfaces are not available. The above I40 requirements cannot therefore be fulfilled.

B. Global networking of process data

For integration of the PLC controllers in supervisor, management and coordination systems (e.g. SCADA or MES systems), which are partly based on web technologies,

The IGF project CICS (18354 N) of the Forschungsvereinigung Elektrotechnik beim ZVEI e.V. – FE, Lyoner Str. 9, D-60528 Frankfurt am Main is funded via the AiF within the framework of the programme for funding industrial community research and development (IGF) of the Federal Ministry of Economics and Technology based on the resolution of the German parliament

978-1-4673-8246-5/16/$31.00 ©2016 IEEE

additional modules are integrated in the PLC controllers, these enabling a bidirectional and event-based process data transmission between the controller and supervisor & management system. This includes, for example, solutions such as the use of Java-Applets on websites for access to Siemens controllers [3], the Web connector with MQTT broker in Bosch-Rexroth controllers [4] or also browser-based access to controllers that already contain an OPC UA server [5].

These solutions also involve proprietary and closed control-integrated modules. Although the modules utilize web technologies, they cannot be transferred to other controllers. The global process data communication is used for HMIs (HMI - Human Machine Interface) and/or supervisor & coordination functions in the higher level of the automation hierarchy (e.g. plant management level). Authoritative statements regarding the time response of the process data transmission are not available. However, different statements result from the latency times of > 100 … 300 ms. Open and consistent web interfaces are not available. The above I40 requirements can only partly be fulfilled with sometimes high adaptation expense for integration in a CPPS.

C. *Introduction of service principles*

Based on the I40 requirement for the service capability of an I40 controller, some projects [6, 7] are involved with the integration of service functions in PLC controllers. Thus the Device Protocol for Web Services (DPWS) enables, as standardized protocol, service-based access to PLC controllers [8] etc. also for reading/writing process data. The internal functional system of a PLC is to be equipped correspondingly for this with the support of the controller manufacturer.

An option for implementation of the DPSW independently of the manufacturer of a PLC is shown in [9]. A service server is implemented here as a functional module based on the standard IEC 61131-3 programming language. This can then be used for the control programming.

However, the DPWS solutions have a principle disadvantage: Instead of reducing or removing the information encapsulation (I40 requirement), further functionalities (service functions) are encapsulated in the controller. Although the service paradigm can consequently be implemented, it significantly encumbers the implementation of other I40 requirements (e.g. Plug&Work, change of dependencies on the runtime). Moreover, DPSW uses the very heavy-duty and complex Microsoft web service protocols. The attainable transmission times of process data via a global network therefore tend to be in the upper range (here too, hardly any definitive statements are available).

In summary, it is to be estimated that there are various solutions and endeavours to provide PLC controllers with additional functions in order to network the controllers in an IP network as defined by Industry 4.0. However, there are considerable deficits in respect to the global networking in particular.

- The process data communication between the controller as a device and a global web functional landscape is slow. Effective statements regarding transmission times and for guaranteeing their quality are not available.
- However, the controllers do not have any web interfaces or, if they do so, such interfaces are inadequately disclosed.
- Available I40 models and architectures have not yet been considered.
- Tools for the migration of classical PLC controllers to the new CPS-based production environment corresponding to Industry 4.0 are practically non-existent.

III. CONCEPT

A. *Basic model*

Initial architecture and reference models have been available for the new CPS-based production structures since 2015. The *Reference Architectural Model Industry 4.0 (RAMI 4.0)* was developed in Germany and published in April 2015 by the ZVEI [10]. Two months later, the *Industrial Internet Reference Architecture (IIRA)* was published by the US organization IIC (Industrial Internet Consortium) [11].

Both architecture models deal comprehensively with the future production process, to some extent from different viewpoints, but essentially remain very general and provide only limited information in respect to a practical implementation. RAMI 4.0 forms an exception in respect to the question of which structure an I40 component should actually exhibit.

According to RAMI 4.0, an I40 component comprises three essential components [12]:

- Physical object (device) or object of the information world (virtual object),
- Management shell and
- Resource manager.

Figure 1 shows the structure of such an I40 component

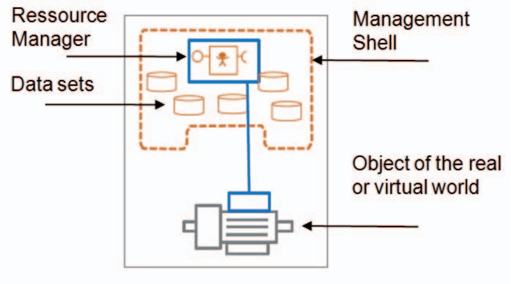

Fig. 1. Component structure of an I40 component according RAMI 4.0 [12]

RAMI 4.0 does not provide any further details regarding implementation of the structure according to Figure 1, nor are any implementations known so far. For the present task of a PLC controller as an I40 component, nevertheless, Figure 1 provides an initial basis that is to be outlined further below.

B. *Device*

The physical object (device) or the object of the information world (virtual object) is only considered generally in the RAMI model. A suitable connection to the IP network is presupposed without further discussion. However, this results in the problem that a normal PLC is only provided with a corresponding web-compatible interface to an inadequate extent. A *Device Gateway* is therefore introduced, which must be able to provide the process data of the PLC controller in the IP network and hence in the web world, rapidly, reliably and securely. The device gateway should be set up so that it can also be integrated in existing PLC controllers without intervention and, if possible, without additional device-technical expense. A “lean” and web-compatible protocol is required for communication with the IP network.

C. *Management shell*

According to RAMI 4.0, the management shell has the following features, amongst others:

- It manages the information belonging to the object in the system environment.
- The management is coherent and consistent in this system environment.
- It has precisely one resource manager.

The management shell is located outside the controller in the information world of the Internet/Web and must therefore also utilize the corresponding technologies from this environment in order to be able to establish a connection with web services, clouds and other objects in the Industrial Internet of Things. In relation to a PLC controller as a device, a *Service Shell* is defined as a functional interface with a management data set, which can be used in the above automation-technical context in a consistent manner (if possible for all devices, not only PLC controllers).

As a giving system environment is assumed for the present solution, the further RAMI property of various management shells does not have to be considered. But also other management shells res. service shells for different system environments are possible (if required).

D. *Resource manager*

The following RAMI properties of the resource manager are of particular interest in relation to a PLC controller as I40 component.

- It organizes the data exchange with the device in online mode.
- It organizes the internal information management.

According to the authors of this paper, the resource manager must also establish a virtual mapping of the device in the selected system environment (Internet/Web) so that it can communicate with other IT objects in the IT world via the service shell. The resource manager is therefore modelled as a *Virtual Device (VD)* for the present solution.

The further RAMI properties of the resource manager such as participants in an I40 service system and organization of the external access to information in the management shell should preferably be assigned to the management shell, in the opinion of the authors.

E. *Model of a PLC as an I40 component*

Based on the above versions, the structure outlined in Figure 2 is derived for the model of a PLC controller as an I40 component.

The communication between the virtual device and device gateway is realized for initialization and parameterization via the HTTP protocol. The websocket protocol is used for rapid bidirectional process data transmission. The pragmatic and “lean” *WOAS Device Protocol* developed in the WOAS project [13] and based on JSON functions is used as an application protocol. In a next version also UBJSON (Universal Binary JSON) will be tested as the data transmission protocol in order to reduce more the roundtrip time.

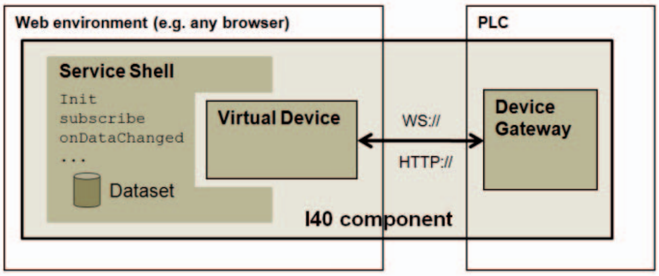

Fig. 2. Model of a PLC as a I40 component

The data set (or multiple data sets) integrated in the service shell contain requisite parameterization and management information for the I40 component. These data sets can be an element of the I40 component itself or outsourced to an external database (e.g. in a cloud).

IV. Implementation

A. *Device gateway*

In line with the specifications for the present concept, the internal structure of the PLC controller is not to be changed, nor is additional device technology such as industrial PCs or specific gateway hardware to be added to the controller. As a result, only the IEC61131-3 control program itself remains as an implementation environment, i.e. a device gateway was implemented as a prototype by means of the IEC61131-3. This device gateway can be executed for the runtime as a functional module together with the PLC control program.

Figure 3 shows the prototype implementation for the device gateway as an FBD program (FBD – Function Block Diagram). The device gateway is integrated as a library with the three functional modules shown in Figure 3 for the PLC programming and can then be used in the PLC program.

Three problems were to be solved for the implementation.

- *Problem 1:* Realization of a websocket server (WS server) is not possible with the standard functional modules of the IEC 61131-3.

- *Problem 2:* For communication with a client, the WS server in the PLC program must know the IP address of the client.
- *Problem 3:* For the process data communication to the virtual device, a protocol converter must be integrated in the device gateway, this converting the utilized transmission protocol (in the prototype the simplified WOAS device protocol) into internal IEC 61131-3 function calls.

Problem 1 was solved by integration of the two open source libraries `OSCAT_BASIC` and `OSCAT_NETWORK`, this providing the Open Source Community for Automation Technology (OSCAT) for various PLC programming systems. The functions required for establishing a WS server e.g. `BASE64_ENCODE_STR.ST` (Convert a string to Base64) and `SHA1_STR.ST` (Calculates the SHA-1 of a string) can be used from these two libraries. It is therefore possible to implement a WS server in FBD language.

Problem 2 was solved by integrating a primitive HTTP web server as a functional module in the device gateway. The web client thus communicates its IP address to the device gateway by means of an HTTP request, which the WS server can then use. An embedded web server that might be present in the PLC cannot generally be used, as this is normally already reserved in the PLC for diagnostic and/or parameterization purposes

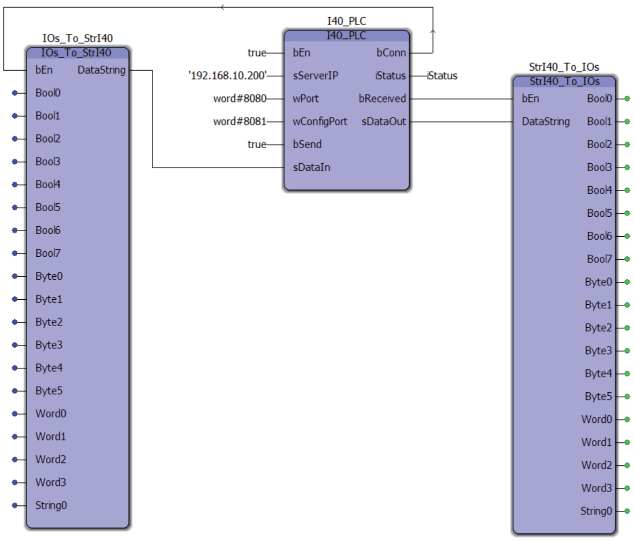

Fig. 3. Prototype implementation for the device gateway as FBD program

A solution to *Problem 3* turned out to be relatively simple, as the standard functional modules available under FBD can be used without any problem for the required protocol conversion.

The HTTP server, the WS server and the protocol conversion are realized jointly in the function block (FB) `I40_PLC`. Two additional FBs are required for the reading and writing connection of the WS server, via which the interconnection with the process variables in the PLC program occurs. The PLC can then provide these process variables in the IP network. This is realized in the two FBs `IOs_To_StrI40` and `StrI40_To_Ios`. The amount of process data to be provided was limited in the prototype for these two FBs. However, both FBs can also be replaced by customer-specific and extended FBs, if required.

B. *Service Shell*

The service shell realizes the management shell provided for an I40 component according to the RAMI 4.0 concept. It is responsible for integration of the PLC controller in the information world of the Internet/Web. At the same time, the service shell therefore also forms the web interface in order to be able to interact with the PLC in regard to its process data.

The service shell manages the relevant instances for the PLC controllers with concrete connection to the system via corresponding parameterized data sets.

The service shell provides a simple and pragmatic set of JavaScript methods for the web client, these serving as a client-end function interface for the PLC controller. Table 1 lists the most important methods for communication between a web client and the PLC as an I40 component.

TABLE I. IMPORTANT METHODS OF THE FUNCTION INTERFACE OF THE SERVICE SHELL

Name	Description
`init`	Initializes the virtual devices (VD) instance on the website, reads all process data of this VD and starts the subscribe method.
`isReady`	This function is called up by the VD if the VD is ready for process data transmission to the real device.
`readChannel`	Asynchronous reading of a process date.
`writeChannel`	Asynchronous writing of a process date.
`subscribe`	Subscription of the VD process data for the event-based transmission.
`unsubscribe`	Stopping the event-based transmission of process data.
`refreshChannelData`	Listing function; called up by the VD object if the value of a subscribed process date is changed.
`getProfilData`	Reading out a VD profile file (if such a file is available)

C. *Virtual device*

The virtual device realizes the resource manager of the I40 component as defined by RAMI 4.0. This involves a JavaScript object that forms a virtual image of the automation device in the Internet/Web world, while communicating via the HTTP or WS protocol. The structure and setup of such a virtual device are described in detail in [14]. Fig. 4 shows the general data model of a virtual device.

Each virtual device consists of a general data as well as specific data. The general data is specified by the manufacturer of the virtual device. The general data characterizes the virtual device class here. The specific data characterizes the virtual device instance for a concrete PLC and is generated and parameterized by a user with the role *Admin*.

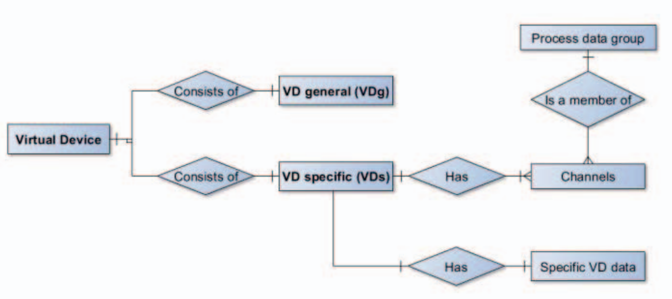

Fig. 4. Data model of a virtual device

A "lean" and flexible device model is required for the I40 component for the rapid and reliable transmission of process data. Familiar device models from automation technology (GSD, FDT/DTM, EDS, FDI etc) are optimized for engineering purposes and less suitable for a runtime operation via IP networks owing to their complexity and large scale. The VD device model therefore uses a simple input/output channel concept (comparable with the port concept from AutomationML) in order to provide process data in a uniform manner for automation services. Each channel has defined parameters and can be extended as required via options (e.g. with a semantic description) A virtual device can have any number of channels.

V. Evaluation and Application

A. Time characteristics

Extensive time measurements were conducted for the PLC as an I40 component. For the client-end determination of the time response, an HTML page generates a pulse frequency that is sent to an output of the PLC. This output is connected via hardware with a PLC input, which feeds its value back to the HTML page. A time difference measurement and logging in a histogram then occurs. This measuring method provides the time characteristics of the bidirectional process data transmission over a longer time period and also logs the dispersion range of the measurements. Figure 5 shows an example of measurements over a time period of 10 minutes for a location of the web client

(a) in the Intranet and

(b) in the Internet (Cologne with cable DSL connection).

The PLC was located in Düsseldorf in both cases.

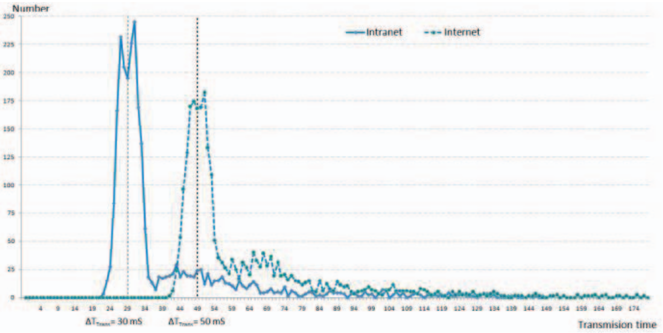

Fig. 5. Time response of a bidirectional process data transmission via the Intranet (a) and Internet (b) with a PLC as I40 component

In the Internet, an average transmission time for the unidirectional transmission of a process datum of **25 ms** resulted when using the I40 component (15 ms in the Intranet). The following approximation can be assumed as a guide value for the transmission time:

$$\Delta T_{Trans} = \Delta T_{Cycle} + \Delta T_{Network}$$

ΔT_{Trans} - Transmission time

ΔT_{Cycle} - Cycle time of the PLC

$\Delta T_{Network}$ - Delay time via the IP network

A similar measurement with a web connector, which works with an OPC DA server as access to the PLC, yielded 40 ms for the average transmission time of the unidirectional transmission of a process datum [15]. The essentially higher transmission time results from the internal delay time of the OPC server here.

A significant improvement to the time response in the IP network is attained by using the I40 component and direct access to the PLC without OPC server.

B. Service integration

An important target from RAMI 4.0 and Industry 4.0 is the application of the service paradigm to automation functions. Realization of the PLC controller as an I40 component provides a uniform interface in the Internet/Web world with the service shell, this able to be used as a service interface.

An example implementation is represented by the CPS Integration Platform WOAS (`http://woas.ccad.eu`). In principle, every PLC can be integrated in this portal as an I40 component when using the described device gateway. Together with other automation services (operating services, visualization services etc.) requisite functional systems (SCADA and HMI systems, remote labs etc.) can easily be created and utilized in the web browser without additional software complexity.

C. Demonstration example

A PLC controller ILC-150 ETH with the programming tool PC WORX from Phoenix Contact is currently used as a demonstration example for the PLC as I40 component. The PLC is integrated in a PLC starter kit with several digital and analog input and output signals as well as peripheral devices (switches, potentiometer, LEDs etc.). Fig. 6 shows the example setup.

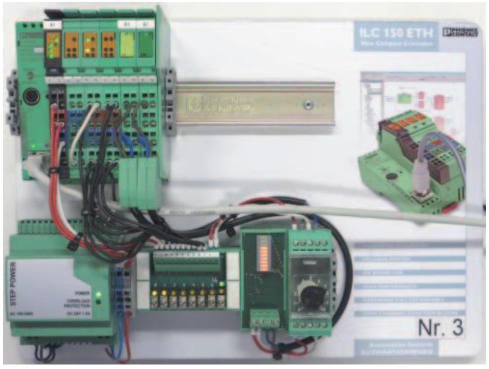

Fig. 6. Demonstration example for the PLC as an I40 component

An example program for operation/visualization of the periphery runs on the PLC. The PLC is already integrated in the above WOAS portal as an I40 component and a publicly accessible website is available for test purposes.

The complete software for the PLC as an I40 component including the documentation and example implementations are available free of charge for downloading and non-commercial use.

VI. Conclusion and Further Works

With the present PLC as an I40 component, a concept was developed with the aim of seamlessly integrating of PLCs in Industry 4.0 product environments using the service paradigm. As a result, previous conventional PLC systems can also be integrated in a new CPS-based res. Industry 4.0 automation without changes.

The concept was evaluated within the framework of a prototypical implementation with the programming tool PC WORX and Phoenix Contact controllers. The fastest possible transmission of process data from a PLC in an IP network is possible with the concept.

Use as an I40 component for controllers from other manufacturers will be examined in the next step. This includes controllers with the runtime system CoDeSys as well as Siemens controllers.

As part of the R&D project "Cloud-based Industrial Control Services – CICS" [16], the use of PLC controllers as an I40 component is also envisaged, in order to outsource non-time-critical parts of control programs in a cloud as services and enable these to execute in the cloud itself and/or on a web client in a runtime machine.

VII. References

[1] *Kagermann, H.; Wahlster, W. and Helbig, J.:* Recommendations for implementing the strategic initiative INDUSTRIE 4.0, Research union: Business - Science, April 2013

[2] *Evans, Z.:* Web Server Technology in Automation. – Siemens Industry, 2011

[3] *Klindt, C.J.; Baker, R.B.:* Interface to a programmable logic controller. - US 6853867 B1, 8. Febr. 2005

[4] *Bosch-Rexroth*: WebConnector: Automatisierungs- und Web-Umgebung einfach verbinden. Product News, 2015, p. 50

[5] *Beckhoff Automation GmbH*: From Sensor to IT Enterprise - Big Data & Analytics in the cloud, ftp://ftp.beckhoff.com/Software/embPC-Control/Solution/Demo-IoT/Flyer-IoT-Sensor_to_Cloud.pdf, 2015

[6] *EU project:* SOCRADES. http://www.socrades.eu, 2006 – 2009.

[7] *Colombo, A. W.; et al:* Industrial Cloud-Based Cyber-Physical Systems. – Springer International Publishing, Swiss, 2014

[8] *Microsoft:* Introducing DPWS. - https://msdn.microsoft.com/en-us/library/dd170125.aspx, 2015

[9] *Mathes, M. et al.*: SOAP4PLC: Web Services for Programable Logic Controller. – 17th Euromicro Inter. Conf. on Parallel, Distributed and Network-based Processing, 2009, Proceedings, pp.210 – 219

[10] *Hankel, M.:* The Reference Architectural Model Industrie 4.0 (RAMI 4.0). – ZVEI, April 2015

[11] *Industrial Internet Consortium*: Industrial Internet Reference Architecture. – Version 1.7, June 2015

[12] *Epple, U.:* Die Entwicklung von Industrie 4.0-Referenzmodellen und Referenzarchitekturen. - Tagung Normung Industrie 4.0 , BMWI Berlin, 19.02.2015

[13] *Langmann, R.; Meyer, L.*: Architecture of a Web-oriented Automation System. – 18th IEEE Int. Conf. on Emerging Technologies & Factory Automation (ETFA 2013), 10. – 13.09.13, Cagliari, Italine, Proceedings

[14] *Competence Center Automation Düsseldorf (CCAD):* Description of a Virtual Device (in German). – Development document, 22.11.13

[15] *Langmann, R.:* HTML5-basierter Web-Connector für OPC. – A&D-Kompendium 2014/2015, publish industry verlag, Munich, pp. 54 - 56.

[16] *CCAD/Fraunhofer ESK:* IEC-61131-based Control Services in the Cloud. – Project flyer, Munich, 2015