Web of Things in an Industrial Environment

Leandro F. Rojas-Peña and Reinhard Langmann
University of Applied Sciences Düsseldorf, Josef-Gockeln-Straße 9, 40470, Düsseldorf, Germany

Keywords: Web of Things, WoT, Websocktes, Web-automation, WOAS, Automation Services.

Abstract: This paper summarizes the work around enhancing an existing Web-oriented Automation System (WOAS) that provides automation services; to support Virtual Devices (VD) that communicate directly with automation components like industrial controllers via Websockets. The presented architecture is a Cyber Physical System (CPS) oriented to production environments in the industry. After a short introduction a quick overview about the currently used technologies is presented. The central part of the work focuses on the implementation of the required interfaces on both sides of the system. The physical side uses a Programmable Logic Controllers (PLCs) as automation device. In the other side JavaScript and PHP are used to develop the client and server modules respectively. A simple demo application was created for the first tests and at the time, some basically delay measurements were realized.

1 INTRODUCTION

In the last years the concepts and paradigms of Web of Things (WoT), Internet of Things (IoT), and Industry 4.0 has been quickly grown and expanded challenging the technology and its applications. At present, we find ourselves at the beginning of a stage, which is characterized by so-called "Cyber-Physical Systems" (CPS). These systems are a consequence of the far-reaching integration of production, sustainability and customer-satisfaction forming the basis of intelligent network systems and processes (Bloem, 2014).

Automation technologies are confronting big challenges due to the increasing penetration of such tendencies. The emphasis on the product portfolio of industrial automation is expected to decline over the next few years in the factory of the future. In contrast, the requirement for globalized service & support is set to gain importance, together with the cost factors. The introduction of secure Cloud computing technology enables access to relevant strategic data via the Internet, which can be used to make decisions in real time as well as boost the operational efficiency (Langmann, 2013).

Many companies and projects has been focus on developments to fulfil the new requirements regarding to the Web-based control applications. Since a few years many alternatives have been used to guarantee the short response time required in such applications using HTTP as standard protocol. HTML-5 brings improvements to the development of Web applications; one of them is the Websockets (WS) specification. WS provide an enormous reduction in unnecessary network traffic and latency compared to other technologies. It accounts for network hazards such as proxies and firewalls, making streaming possible over any connection, and with the ability to support upstream and downstream communications over a single connection. HTML5 WebSockets-based applications place less burdens on servers, allowing existing machines to support more concurrent connections (Kaazing). This makes it suitable for many applications requiring fast and frequent action in both direction client/server/client; like those which involve multiple users communicating with each other, or those where the server-side data constantly change (Freeman, 2013).

As consequence, Ethernet is nowadays used as a standard protocol for the factory floor and other industrial applications. But it is important to note that the timing is a quite critical in the industrial environments networks. "Time critical" refers to a system's dependency on a deterministic latency - or better, absolute knowledge of time to a very high degree of accuracy. This is the major difference between information exchange and control applications (Zarr, 2014).

The main objective of this development was to upgrade the WOAS system to support Virtual De-

F. Rojas-Peña L. and Langmann R..
Web of Things in an Industrial Environment.
DOI: 10.5220/0005491105330539
In *Proceedings of the 11th International Conference on Web Information Systems and Technologies* (WEBIST-2015), pages 533-539
ISBN: 978-989-758-106-9

vices that communicate directly with automation components like PLCs via Websockets. The test and demo scenarios as well a simple real application were also part of the initial objectives.

2 RELATED TOPICS

This section will briefly introduce some related concepts and technologies and its state of the art.

2.1 Trends on the Nowadays Web Automation

Nowadays there are several topics that have been the subject of numerous articles focused on this field. These include Internet and Web of Things (IoT, WoT), Big Data, Cloud Computing and Industry 4.0. All of these trends involve a lot of devices networked together and a lot of data available to do things. They also include deciding whether data is stored and applications accessed from the computer next to you or from a server located somewhere else (Oulton, 2014).

Industrial automation spans a wide range of control systems from simple Programmable Logic Controllers (PLC) to Supervisory Control and Data-Acquisition (SCADA) systems and distributed control systems (Joby, et al., 2011). Automation systems frequently control multiple devices in a synchronized fashion. PLCs generally are end nodes for more complex control systems. They are simple to program and these days; PLCs are microcontroller-based systems that may have a procedural programming interface. In any case, a typical PLC will offer rugged I/O and real-time synchronization. Today's PLCs may have network connectivity as well. A network does not affect the I/O side, but timing and synchronization will be an issue (Wong, 2014).

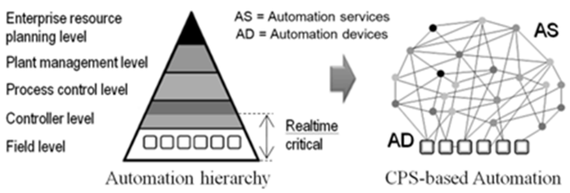

Figure 1: Decomposition of the automation hierarchy by CPS with distributed services.

The conventional automation hierarchy is increasingly transforming into a flat automation "Cloud" as Figure 1 shows. Service-oriented architectures (SOA) provide a good solution for creating uniform interfaces, while allowing collaboration from the field level to the enterprise level. Cloud computing is the provision of jointly usable and flexibly scalable IT services using IT resources via networks. Typical features are provision in real time as a self-service based on Internet technologies and charges according to use. Cloud computing enables users to redistribute from investment to operational expense, while paving the way for completely new business models and their prompt implementation (Langmann, 2014).

Current automation systems such as process visualization systems, SCADA/HMI systems and process control systems are proprietary and manufacturer-specific systems that are created with conventional programming languages and mostly have manufacturer-specific internal interfaces and data formats, making them hardly to accept as an open industrial model. Also the increasing standardization and opening of interfaces and description methods (OPC UA, FDT/DTM etc.) does not change the fact of the unity of an automation system, but complement them only in the direction of openness of a given environment. Not the systems themselves are more open, but rather only the interfaces and description methods are more standardized and opened. Only the new and standardized IT technologies with Web/Internet at the forefront established the prerequisites for a spatially unlimited distribution of functions/services and are now also increasingly interesting for automation technology (Langmann, 2013).

2.2 WOAS

In general all functions which are required for the execution of an automation system could be used as services. These can be such simple functions as a measuring algorithm, operating or visualization of process data, but also complex functions from higher levels (Figure 1) could be used as AS. Looking at an automation system from this perspective there is still some open points; the new automation structure is a vision for the future. At present, no application-wide systems are available on the market, which implement a CPS-based and service-oriented automation.

Between 2011 and 2014, however, the prototype of a Web-based platform was created, which can integrate worldwide distributed services also with worldwide distributed CPS components (device components) into a functional system (Langmann, 2013). This was within the framework of the R&D project "Architecture and Interfaces of a Web-Oriented Automation System (WOAS)" and involved the participation of 10 industrial companies.

The WOAS platform can also be referred to as CPS integration platform. On the client side (browser) the platform is completely implemented in HTML5 and JavaScript runnable in each browser and device.

For the CPS-based automation, the automation devices (AD) like controls, sensors, actuators etc., can be considered as CPS components after adding corresponding interfaces to the IP network. This results in the CPS structures as shown in Figure 2 for different AD. A Virtual Device (VD), which maps the real device in the virtual world, functions as an interface from the CPS components to services distributed in the network (Langmann, 2014, p. 133).

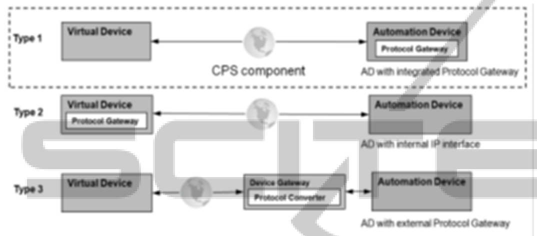

Figure 2: Automation Devices as a CPS components.

The VD is used to map the process data of an automation device via event-based channels onto Web- or Internet-suitable objects in a uniform manner and makes this available in a Web browser. Via integrated protocol or device gateways, any industrial interfaces can therefore be made available in the IP network. For the process data transfer between a VD and an AD via HTTP, a specific protocol was developed. The VD and the protocol- or device-gateway together form a Web Connector for the AD as a CPS interface.

This work describes the development and evaluation of an integrated PLC Web Connector according to a Type 1 VD (Figure 2) and its application in the WOAS portal for creating Web-oriented automation systems.

2.3 The WSS4ILC Library

Even when the Websocket (WS) protocol is not anymore so young in the world of the standard Web applications, it is so in the automation world, especially in the industrial controllers like PLCs. All the more, most automation manufacturers provide a Web server in their devices, but usually it is only for managing purposes or simply do not support WS. To be able to explode the benefits of the WS protocol in the automation devices it is necessary its complete implementation.

Fortunately, for the PLCs of the Phoenix Contact family (ILC series) exist a library that provides a basis implementation of the WS protocol which main technical specifications are: data frames can be up to 125 byte long; support the Ping/ Pong signals and last one is a kind of limitation but enough for this application; it allows only one client at a time.

The use of the library is a quite simple and requires just to instantiate the WSS4ILC FB and set or connects all input and output parameters properly (Rojas-Peña, 2014).

3 SCIENTIFIC-TECHNICAL CONTRIBUTION

Once covered a few obligatory topics around the concepts related to this work it is possible to present its global conceptual idea. Figure 3 shows how industrial controllers or any automation device in general can be directly connected to the Cloud allowing its control and/or supervision by any client user using an Internet browser. Any user can access the WOAS portal, but to access the services provided by the controller as a VD must be properly register in the WOAS server and be subscribed to the desired channels.

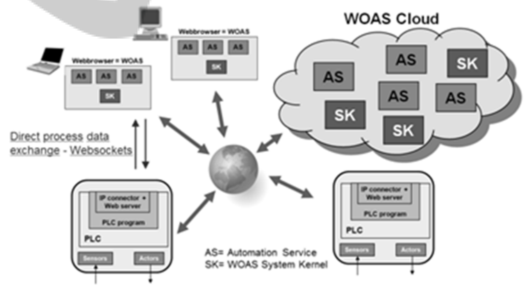

Figure 3: Conceptual idea.

3.1 The VD4PLC Concept

To be able to successfully fulfil the proposed objectives it is necessary to properly understand the complete structure of the architecture and its functionalities. Figure 4 shows the connections between the different parts of the automation system and the used protocols. Note this corresponds to a single automation device, in this case, the industrial controller used for the test, an ILC 150 ETH PLC. In this sense, VD4PLC is the VD developed to allow WOAS connects directly with industrial controllers.

Let's take a closer look to each part of the system and decide how to realize its implementation. In one side is the automation device (AD) and in the other side the Virtual Device (VD) which is a JavaScript dynamic Web page created by the WOAS system kernel. Once a user logs in to the WOAS client, an instance of the registered VDs is created. In this process a WS communication is established between both entities allowing the bidirectional data exchange. As a particularity for PLCs must be considered that the programming standard IEC 61131-3 do not allows using the device's Ethernet interface with unknown entities. For this reason it was necessary to find a way how to set the client's IP that is starting a session in the PLC.

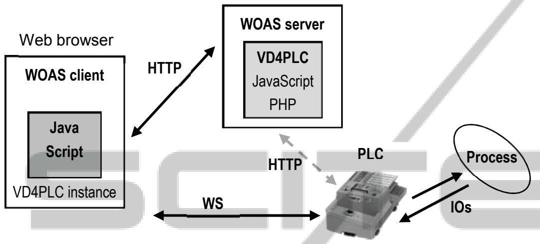

Figure 4: Structure of the system.

3.2 VD4PLC Interfaces Implementation

To communicate both sides of the system correctly it is necessary not only to use the WS protocol for the communications between the PLC and the VD in the WOAS client but also to implement a higher level protocol to properly perform the data exchange according to the WOAS specifications.

To be congruent with WOAS it is necessary to process the commands *Subscribe*, *Unsubscribe*, *Read* and *Write* (CCAD-VD, 2013) as they will be sent by the WOAS server, or a test client during the development phase. This process is implemented very simple; the first byte in the sent/received data string is a command field and codifies this functionality as Table 1 shows.

Table 1: WOAS commands.

Command	Description
Subscribe	Data can be sent when they change or be explicitly requested
Unsubscribe	No data will be sent (stops sending)
Read	Explicit data update requested. A frame should be sent with current value of one or all subscribed channels
Write	Contains the updated/new values of one or all subscribed channels

Also note that the frames with the commands *Subscribe*, *Unsubscribe* and *Read* are considered command frame's that contains no process data, while frames with de command *Write* are considered as data frames and should contain process data.

With the use of the library WSS4ILC the heavy work is already guaranteed in the side of the AD because it provides the required WS server as basic unit for the communication. It is only necessary to implement the protocol previously described in the AD or PLC and to interact with the process I/O as the automation function requires.

The WOAS protocol was programed as a modular function block (FB), which in correspondence with the WSS4PLC, use parameters for its configuration, facilitating its portability. The used development environment was PC WORX from the Phoenix Contact company. At this point the interface with the WOAS VD is ready to be used.

In the client side, the VD4PLC is programmed in JavaScript and of course implement the above described protocol as the other edge of the communication channel. Its main module is the file `vd4plc.js`, including functions for sending and processing messages to or from the industrial controller. Those functions are:

- *sendToPLC*: Prepares the messages (strings) to be sent to the PLC when commands are generated. Write commands with process data; the others without.
- *getFromPLC*: Decode the messages (strings) received from the PLC when an explicit read command is generated or the automation device sends new process data automatically due to its own functionality or data updates.

A third function has an important role in this main module; it is the initialization function which by means of Ajax requests to the WOAS server setting the client IP in the PLC.

An IEC 61131-3 programmable industrial controller needs to know the IP of its partner before starting a connection. To overcome this problematic or limitation it is necessary to fetch the client IP at the moment the user logs in to the WOAS portal and send it to the PLC. After, the WS library can connect with the VD allowing direct data exchange between the user somewhere in the Cloud and the automation device (AD) in a certain physical process. Two things must be done in this direction; the VD4PLC also includes a PHP module (`vd4plc.php`) which at the proper time fetches the client IP and sends it to the PLC (function *sendCmdToPLC*) via HTTP. Therefore the AD also requires a Web server; it was also part of this development and with quite rudimentary functionalities allows setting the IP for the proper WS communication and also to know the stay of the WS server in the WSS4ILC library.

4 TEST-DEMO APPLICATION

For the initial test a Phoenix Contact EduNet development kit was used as an automation device. This kit has a PLC of the low range ILC (150 ETH) series of the same company. The digital inputs (DIs) are connected to a block of switchers allowing its manual operation. The analog input is connected to a linear potentiometer and the analog output to a led bar with 12 indications levels. To have a direct feedback of the hardware functionalities one of the DOs is physically connected to one of the DI's.

A simple program was written to combine the WSS4PLC library, the developed WOAS FB written in the IEC 61131-3 programming language and a dummy automation function that basically forward the values of the inputs every time they change and in the other direction reflex the received values to the outputs. The idea was to simply show how to create automation services that will be available for the WOAS users in the Cloud.

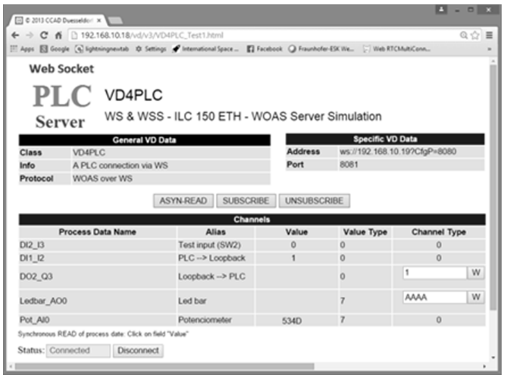

Figure 5: Test Web page.

To test the WOAS client was used a standalone Web page. This is possible because WOAS is complete modular and the VD4PLC implementation can be used out of it. This has a big advantage allowing control automation developers to easily access its process from the Web. The unique requirement is to use Web server to host the PHP component of the VD.

The Web page used as user interface to test the developed system is a quite simple, see Figure 5. Note some of the VD properties are shown in the upper tables while the data channels equivalent to the automation services provided by WOAS are in the lower table. The two controls at the bottom facilitate the tests; a *Status* field indicating if the WS is connected to the server or not and a *Disconnect* button to close the connection explicitly.

In the middle there are three buttons which are equivalent to the WOAS commands *Read*, *Subscribe* and *Unsubscribe* respectively. Note that the *Read* command is also triggered with a mouse click over the channel Value column in the bottom table. For the *Write* command it is necessary to click over the button ‘W’ in the Channel Type column.

During the days of WEBIST‘2015 the link http://vd4plc.ccad.eu will be available for online tests with this demonstration application.

5 DELAY TEST

A delay test was performed to have an idea about the time performance of the implemented system. This test was realized in two different environments, the first one in local closed scenario in intranet and a second one in an open scenario in the Internet. In both cases was used the same web page on the PC and the same program on the PLC. The physical measurements were based on the loopback connection of one output and one input on the automation device which guarantees that certain information received will be automatically echoed. Around 2500 messages (pulses) were sent; while some values were computed: maximum and minimum delay, total of pulses lost (no pulse confirmation received), and the frequency distribution of the delay according to the number of pulses with a same delay.

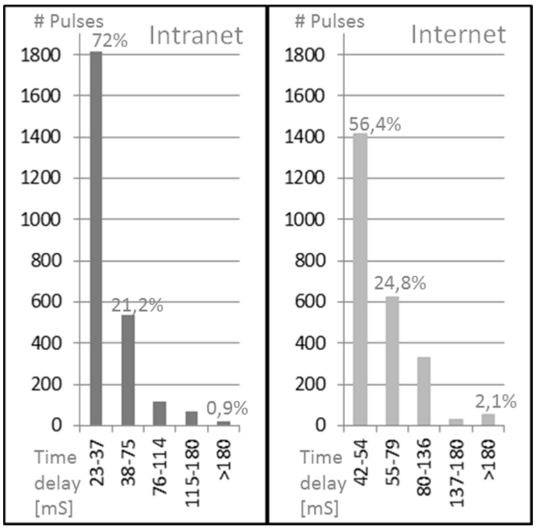

Figure 6: Delay analisys.

Figure 6 shows the results of both tests graph ically in different interesting time windows according to the delay range in the abscise axis. The blue curve (left) represents the obtained values for the test realized in the intranet with a very good time response. Most measured delays were between 23 and 37 milliseconds for a 72% of the total sent pulses. The minimum value is really small and can be considered ideal for some applications. The maximum value is still acceptable for some applications but has not so important significance because it is just a punctual value.

The green curve (right) depict the values for the test realized in the Internet with bigger delays and also minor repetitions for smaller delays but also acceptable for the used scenario. In this case, most of the values were between 42 and 54 milliseconds for a 56,4% of the total sent pulses. Here is also interesting to note that between 55 and 79 milliseconds there is an important amount of values equivalent to the 24,8% of the total sent pulses. The minimum value is not so small like in the intranet test but can be also considered ideal for some applications. The maximum value is much bigger than in the other test and should be considered significant depending on the application to be used.

In both cases the values are considered very well for “soft” real time applications in automation (Home, 2014). The percent of delay values above 180 ms is only 0,9% in the intranet test and a 2.1% in the Internet test. The lost pulses in both cases are around 0.2 percent.

6 CONCLUSIONS AND FUTURE WORK

The present work describes the implementation and test of an automation system which is services oriented and CPS-based to be used in the WoT as an enhancement of the WOAS project.

The new developed Web connector for the PLC (Virtual Device and WOAS Function Block) allows to connect automation devices directly to the Cloud, and therefore to be controlled and supervised by a user with the valid and with the required rights from the Cloud. The tests realized with a demo application show the feasibility of the implementation to be used in the production environments in the industry. On the other hand the time measurements results are very promising in comparison with previously technologies, suggesting this architecture to be used in control automation Web-based applications where the time constraints are considered soft with relative small jitter on medium-low network's load.

There are still a lot to do in this direction. The first steps are directed to allow the WSS to support more clients simultaneously and at the same time improve the WOAS protocol designed for the VD4PLC allowing longer frames and better handling of the modified or read data. As a generalization both, the WSS library and the VD should be extended to other types of industrial controllers with the respective tests.

Cloud-based Industrial Control Services (CICS) is a new R&D project focused to a complete Web-oriented control system (see control level on Fig.1). Industrial control programs according the standard IEC 61131-3 should be executed as control services from the Cloud. The described WoT solution for PLC will be evaluated in the CICS project as one of the possibilities for a direct connection of PLCs as CPS component to a Cloud-based and distributed control system.

REFERENCES

Bloem, J., 2014. *The 4I Revolution: Industrial – Information Tech – Internet – and Industrious Once More.* s.l.:SogetiLabs.

CCAD-VD, C. C. A. D., 2013. *Beschreibung der Virtual Devices.* s.l.:s.n.

Freeman, J., 2013. *9 killer uses for WebSockets.* s.l.:InfoWorld.

Home, S. P. S. D. C., 2014. *Design of communication systems for networked control system running on PROFINET.* WFCS 2014, Toulouse: IEEE.

Joby, A., Mahato, B., Sharma, S. & G., C., 2011. *A web PLC using distributed web servers for data.* ICISA 2011, Jeju Island: IEEE.

Kaazing, C., n.d. *About HTML5 WebSockets.* s.l.:WebSocket.org.

Langmann, R., 2014. *Automation services from the cloud.* Industrial Ethernet Book: s.n.

Langmann, R., 2014. *Ein Interface für CPS-basierte Automatisierungsgeräte (Interface for CPS-based automation devices).* AALE 2014: Deutscher Industrieverlag.

Langmann, R. M. L., 2013. *Architecture of a Web-oriented Automation System.* ETFA 2013, Cagliari/Italy: IEEE.

Oulton, B., 2014. *4 Big Trends that Impact Industrial Automation and What To Do About Them.* s.l.:Belden.

Rojas-Peña, L., 2014. *Making your PC WORX project WOAS compatible,* Düsseldorf: CCAD.

Wong, W., 2014. *Electronic Design: Industrial Automation Relies On Ethernet.* s.l.:electronic design.

Zarr, R. F., 2014. *Electronic Design: Ethernet Dominates Industrial Environments.* s.l.:electronic design.

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.