

Distributed Web-Based Control System

[doi:10.3991/ijoe.v6i3.1345](https://doi.org/10.3991/ijoe.v6i3.1345)

Reinhard Langmann

Duesseldorf University of Applied Sciences, Duesseldorf, Germany

Abstract—This article describes a concept and application examples for a distributed Web-based control system (DWCS). The DWCS is based of two key components: an IEC 61131-programmable Web control and a process data proxy as the process interface. Control functions can be distributed and executed ad lib in the Intranet/Internet via the DWCS.

Index Terms—Distributed Web control, Web-based control, DWCS, distributed control system, process data proxy, lean Web automation

I. OBJECTIVE AND STATE OF THE ART

The classic definition of a distributed control system (DCS) is a control system with control elements (controller) that are not grouped centrally but rather distributed in the system, with each sub-system component being controlled by one or more controllers. This definition is based on the assumption that each component to be controlled (equipment, machine, device) actually includes a physical device controller, like a PLC for instance. Most publications in the field of distributed automation systems address such systems; in particular with regards to an efficient communication between controls distributed in a network and the simulation of the overall system (see also examples in [1, 2, 3]).

The actual benefits of a distributed system, like e.g. failure and error tolerance, transparency, scalability and openness can hardly be used as a DCS is based on the control device component and in substance mainly focuses on the device distribution and communication between the devices.

Another group of publications therefore deals with the distribution of the control functionalities (control algorithms) in a distributed system, where the individual equipment and machines to be controlled do no longer require controllers as device components. The required control functions can be allocated to N controller (computer) nodes in the distributed system (network). The equipment components contain suitable interfaces (process data servers) only, which provide the required process signals in the distributed system.

Such solution for distributing control functions based on Java is described e.g. in [4]. According to this solution, distributed Java applications provide the control functions. The equipment to be controlled is connected through a field bus via a field bus server. This server is equipped with a Java remote interface. RMI (Remote Method Invocation) is used for the communication between the Java applications via the Intranet/Internet. The exclusive fixation to a pure Java system requiring also the development of the control programs in Java is a drawback, as this can hardly be conveyed for industrial applications.

An interesting approach for distributed control is described in [5] on the basis of function proxies. A function proxy serves as a proxy for distributed functions and is implemented using the industry standard IEC 61131. They allow the creation of distributed systems for heterogeneous automation products.

Previous publications have only scarcely considered the current Internet and Web technologies that are already available as a distributed system and can also be used for control-technical applications.

From the field of business applications in relation with the use of Web technology there are service-oriented architecture (SOA) which support various services on servers in an IP network [6]. A user can use these services of any client (Web browser). As Web technology, Web services are used with SOAP/XML and the HTTP protocol.

To be used for automation systems, there are a few suggestions ([7, 8, 9]) from the research. However, these solutions are based on Enterprise Web Portal technologies and create very complex and weighty systems. The application of these solutions for the distribution of control functions is not known.

In the EU project SOCRADES [2] a consortium of leading automation companies is involved in the development of network-distributed automation systems based on a SOA structure. The focus here is also almost exclusively on the preferred option of Microsoft technologies (Web services with SOAP/XML), and so there are also the aforementioned problems (s. also [10]). Moreover, the focus in this project is more on visualization, diagnosis and data management, and not on the distributed control.

To the state of the art belong further works, which deal itself with the use of XML for different applications in automation [11]. XML is used thereby increasingly as universal meta language, in order to e.g. describe device models, protocols or configuration files. Also the description of more complex systems e.g. for the maintenance of automation plants can take place with XML [12]. A use of XML as basis for the Web-based distribution of control functions is so far however not known.

The subject of the present publications is the development and application of a Distributed Web-based Control System (DWCS), in which the control functions are realised through Web objects distributed in the Intranet/Internet. The control programs shall be developed with conventional engineering tools, like e.g. with the IEC 61131-3 programming environment PC WorX (Phoenix Contact).

II. SCIENTIFIC TECHNICAL SOLUTION

A. Concept of a Web-distributed control

The following four basic structures can be defined based on the key options for the distribution of control functions as Web-based control objects¹:

- A Web-based control object controls one single device connected to the network (1:1 control).
- N network-distributed control objects control a single device connected to the network (N:1 control).
- A network-based control object controls N devices that are connected via the network (1:N control).
- N network-distributed control objects control M network-connected devices (N:M control).

Each basic structure offers specific benefits based on the application scenario. Chapter III provides a detailed description of some test applications and their characteristics.

The following tasks must be completed for a DWCS under consideration of the client-server principle applying for the Web:

1. Development of a generic control object that is capable of executing industry-compliant IEC 61131-3 control programs on various client computers.
2. Development of a process data interface for the technical equipment to be controlled in order to provide the required process data in the Web in a timely and flexible manner.

The solutions should be designed so as to enable the computers used for Web-based control (control nodes) to load and activate the required control objects and process data interfaces via a Web browser as required.

B. Programmable Web controller as a key element

A control object designated as a “Programmable Web Controller” (PWC) as a reference to a PLC is the key element of the DWCS [13]. This object is both capable of processing any PLCopen-XML program and interaction with other objects on a HTML page.

A PLCopen-XML program is the XML version of a PLC program standardised by the PLCopen consortium (Fig. 1). Industry standard PLC projecting tools, like e.g. PC WorX or CoDeSys enable converting traditional PLC programs into PLCopen-XML programs and vice versa via export/import functions.

The PWC operates as a runtime environment for a PLCopen-XML control program, enabling the control to be programmed vendor-independently in the industry-standard notation with known tools. The PWC object can generally be realised through any programmable Web object. However, the PWC was developed as a Java applet for security reasons.

In addition to the runtime engine, the PWC applet of the DWCS is also equipped with a graphical user interface (GUI) that can be activated for diagnostics purposes and the appearance of which resembles to a modular PLC (Fig. 2).

Public applet functions and the LiveConnect mechanism enable other objects, like e.g. process data objects to communicate with the PWC via JavaScript. Table I lists the implemented public PWC functions in this context.

The PLCopen control program can be stored on any computer node in a network (even on the client computer) and is subsequently loaded into the PWC and executed via the loadPWC function. The current release already includes most of the standard operations, functions and function elements according to the IEC 61131-3 standard in the PWC.

The PWC applet is configured and the control features are set through a parameter set. This parameter set includes e.g.:

- The number of digital IN modules (8 INs each);
- The number of digital OUT modules (8 OUTs each);
- The number of analogue input/output modules (with two inputs/outputs each);
- The cycle time of processing the control program in ms;
- Source address of the PLCopen-XML program (for direct load).

```

...
- <body>
- <IL>
- <html xmlns="http://www.w3.org/1999/xhtml">
- <p xmlns="http://www.w3.org/1999/xhtml" xml:space="preserve">
LD hand
<br />
ANDN auto
<br />
ANDN step
<br />
ST m_hand
...
</p>
</html>
</IL>
</body>
...
    
```

Figure 1. Excerpt of a PLCopen-XML program

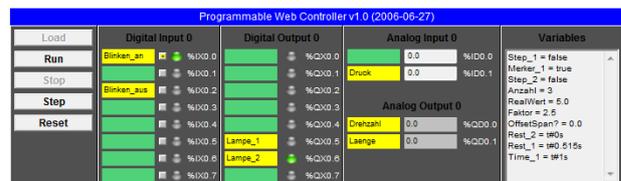


Figure 2. PWC diagnostics interface

TABLE I. LIST OF PWC FUNCTIONS

JS functions calling by the PWC applet	
Function name	Description
[appletID]_onInit ()	Call after initialisation
[appletID]_onProgramLoaded ()	Call after loading a PLCopen-XML program
[appletID]_onDataChanged (changes)	Call after changing on an output
[appletID]_onError ()	Call after an error during the program processing
Public funtions of the PWC applet	
Function name	Description
loadPWC ()	Load a PLCopen-XML program
runPWC ()	Start the control program
stopPWC ()	Stop the control program
resetPWC ()	Reset the controller
readPD (String pd)	Read a process data
writePD (String pd, String value)	Write a process data

¹ The “Web” in this case refers to an Internet-protocol-based network, irrespectively of whether it is a closed Intranet or the open Internet.

Fig. 3 shows the integration of a PWC applet into an HTML page.

The control programs for the PWC are developed with industry-standard programming tools and exported as an XML program. In contrast to the otherwise common IEC 61131-3 programming with symbolic addresses, absolute addresses must be used for the PWC programs in order to be able to allocate the input/output addresses in the control program to the inputs and outputs of the PWC.

C. Process interface

The connection of the network-distributed PWCs to the respective process signals of the equipment to be controlled is established using the *Lean Web Automation* concept (LWA) [14]. This approach enables an event- and requirement-based provision of sensor and actor data from automated equipment in the Intranet/Internet. The process data are transferred using the Java-based *W2<process>-Proxy* application model via LiveConnect to this end. This model provides the PWC with an interface to the process data of a remote equipment similar to a proxy (Fig. 4).

The *<process>* placeholder in the application model represents the respective practical realisation of the corresponding process data interface; e.g. to an OPC server, a driver for a field bus system or an embedded controller. The *W2<process>* proxy server is a stand-alone application that is executed on a process data server (embedded in the equipment or on a server computer).

The following proxies are currently available:

- W2OPC proxy: Interface to any OPC servers,
- W2WebIO proxy: Interface to the embedded Web-IO of Wiesmann & Theis (www.wut.de),
- W2BeckIO: Interface to the embedded web controllers of BeckIPC (www.beckipc.de).

The *W2<process>* proxy client establishes a permanent dynamic data connection to the *W2<process>* proxy server in the process data server and – via the connection to the process data interface – to the actual equipment process data. The data connection between the *W2<process>* proxy client and the server uses a TCP socket connection. The process data are transferred according to the subscriber/publisher principle using a simple command protocol.

The PWC functions work as a process data service (PD service) according to the LWA principle and can be stored on any server computer (PD service server) in the network.

The connection between the *W2<process>* proxy applet and the PWC applet is established via JavaScript on the respective HTML page. An unlimited number of PWC can interact with any number of process data proxies on each HTML page. This ensures an easy and flexible distribution of control functionalities to the equipment of various manufacturers with different process data interfaces.

III. APPLICATIONS

Based on the developed DWCS, control functions can now be distributed and activated almost ad lib in the network, depending on the actual requirements, in order to realise distributed Web-based control and automation systems.

A. Control of a mechatronic equipment

A first function example describes the operation of a processing and test station (station) from the MPS program by Festo Didactic (Fig. 5) via DWCS according to the 1:1 control principle.

The station includes a total of 14 digital I/O signals and is directly connected to the network via a WebIO module [15]. The WebIO module is equipped with a Web server and provides its process data on per-event basis via a process data server to TCP/IP.

A *W2WebIO* proxy applet is loaded from this Web server to a HTML page with a PWC applet and establishes the connection between the control and the I/O signals. The component structure of this example corresponds to the general structure illustrated in Fig. 4.

The control program performs the following functions:

- Operating mode selection: Manual/automatic/step,
- Manual operation for testing the station’s actors,
- Automatic operation for a fully automated sequence of the overall 11 processing steps,
- Step operation for performing individual processing steps.

```

...
<APPLET archive="pwc10.jar" code="pwc.PWC.class" name="pwc"
height="210" width="700" mayscript="true">
  <PARAM name="appletID" value="PWC">
  <PARAM name="digInUnits" value="1">
  <PARAM name="digOutUnits" value="1">
  <PARAM name="realIOUnits" value="1">
  <PARAM name="source" value="html">
  <PARAM name="xml" value="">
  <PARAM name="period" value="10">
</APPLET>
...

```

Figure 3. Integration of the PWC applet into an HTML page

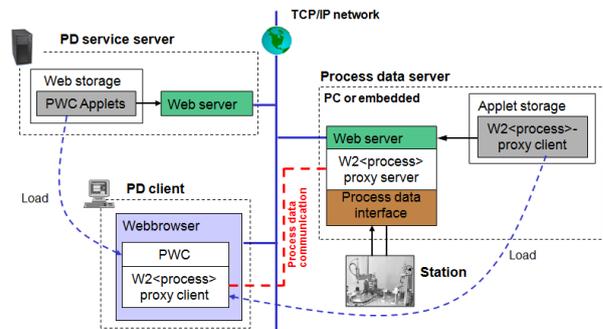


Figure 4. LWA structure using the example of a distributed PWC control



Figure 5. Processing and test station

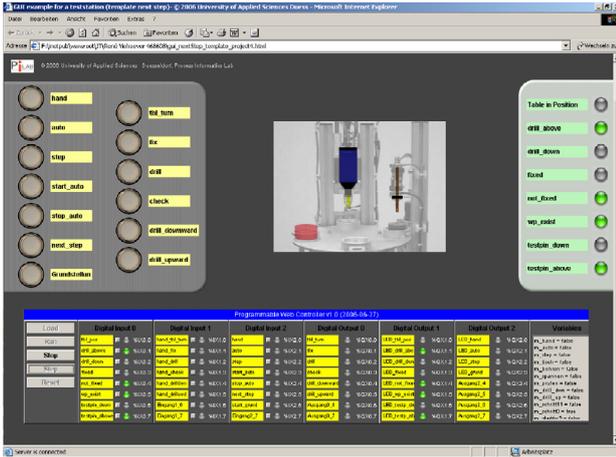


Figure 6. User interface of the DWCS for controlling the processing and test station according to Fig. 5

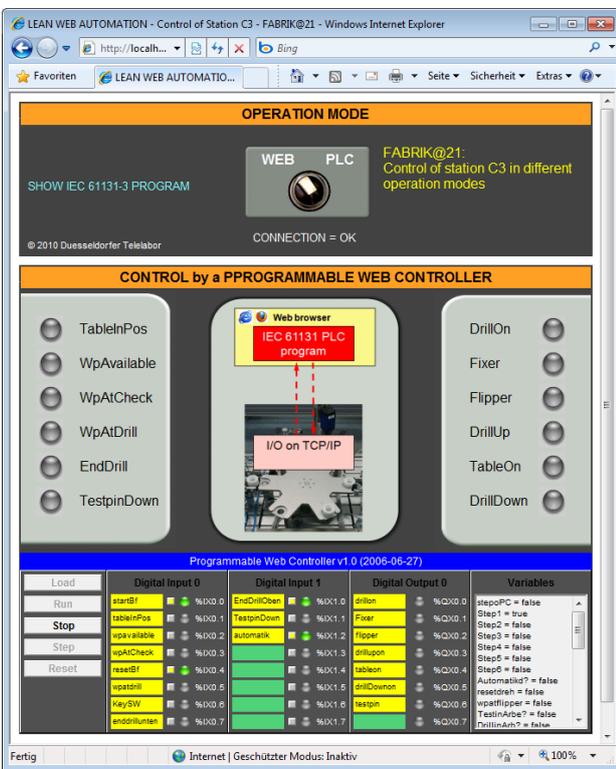


Figure 7. User interface of the SIL simulation for controlling the processing station C3 in the FABRIK@21

The programming and export into a PLCopen-XML program is performed through the PC WorX PLC programming system. After the export into the XML program, the control program includes 864 lines of XML program code.

The station is operated and visualised through an additional HTML operating user interface. The PWC diagnostics interface is activated in addition for further control. Fig. 6 shows the presentation in the PD client, which now serves as the control computer for the station.

While the DWCS usually works seamlessly in the Intranet and Internet, remember that this is not a time-critical system: a delay of individual operating steps (e.g. network delay times) only extends the duration of the overall processing cycle.

In a further example according to the 1:1 control principle, the DWCS is used for a software-in-the-loop simulation (SiL) of a processing station in the model factory FABRIK@21 [16] (Fig. 7).

Thereby, can be switched by means of a Web page simply between the processing of the control program in the hardware PLC or in the Web-based PWC. In combination with a Web-based simulation of the mechatronics as a 3D-Modell (hardware-in-the-loop simulation - HiL) the physical real stations of the model factory can be shifted flexibly and Web-controlled into any client PCs as stations of a "Digital Factory". Thus various possibilities of a mixing process with real stations and in the Web distributed virtual stations can be used for test, learning, start-up and diagnostic purposes.

B. Reliable distributed control

Ensuring reliability is a major problem when using distributed control functions in a network, as the very network infrastructures must be considered when determining the reliability. In particular when operating the system in the Internet (Web), potential unforeseen delays and network failures can occur. This issue brings up the question of how a DWCS can be operated reliably via the Internet. The Pi-LAB analyses solutions that increase the statistical reliability by using a control swarm to this end.

The control swarm principle features an operation of an automated equipment by N parallelly switched redundant PWCs in the network (N:1 control). Due to their Web technology base, numerous redundant PWCs can be easily distributed on a multitude of individual computer nodes through HTML pages, making temporary failures of individual PWCs on individual computer nodes tolerable due to the high number of parallel controls.

C. Control of constant energy flows for distributed power plants

Using DWCS provides a particular advantage for non-time-critical controls of equipment distributed at large distances. This includes e.g. wind power plants, oil mining stations and distributed power plants. Multiple units within a group (e.g. wind power plant) can be easily controlled according to the 1:N control principle through a single superordinate PWC with regards to a group parameter (energy flux of the wind power plant, output per mining field, etc.).

D. Distributed equipment control

The DWCS flexibility can be maximised by controlling N devices and/or components through M PWCs. This approach offers advantages in the following cases:

- Geographically distributed equipment components are characterised by an increased functional dependency and must be synchronised extensively. Controlling these components through a single PWC eliminates external synchronisation efforts, as all equipment components are operated through a single control program.
- Some functional groups are distributed over multiple equipment components. A PWC assigned to this group can control this functional group easily and transparently.

Such a DWCS is currently built for the FABRIK@21 [16] operated at the Pi-LAB in order to demonstrate the feasibility and the benefits.

The FABRIK@21 is a model factory for hybrid production processes, which manufactures a fluid that is filled in bottles and stored in sixpacks. The factory consists of 19 substations. In a first step, three substations for processing bottle caps will be controlled by four PWCs distributed in the Intranet. In principle for this purpose the represented PWC Web page in Fig. 7 will be quadrupled and the control functions will be distributed on three stations.

IV. CONCLUSION AND FUTURE WORKS

The article describes the principle of a Web-based distributed control system (DWCS) enabling the distribution of control functions for automated equipment in the Intranet/Internet, using standard Web technologies. The article explains the two key DWC components: the programmable Web control (PWC) and the process data interface according to the proxy principle. Test applications already completed and potential application examples illustrate the distributed control principle.

The presented DWCS has following advantages among others:

- Necessary proprietary and manufacturer-specific PLCs as hardware can be omitted and can be shifted for non-time-critical tasks as software into the IP network.
- Control functions can be used and managed flexibly as services. Thus new business and application models can be developed (e.g. "rent" a control function).
- The DWCS supports transparency, scalability and openness of distributed control functions and makes a flexible SiL simulation possible for the „Digital Factory“.

Future works will focus on solutions designed to ensure the DWCS reliability by using a control swarm and on extending the application examples in order to demonstrate the benefits of this concept.

To the further scientific-methodical extension the described DWCS is embedded in addition into a superordinate new project "Architecture and interfaces for a Web-oriented automation system (WOAS)", in which 10 well-known German automation engineering companies participate.

REFERENCES

[1] H. Wurmus, „CNet - Komponentenbasierter Entwurf verteilter Steuerungssysteme mit Petri-Netzen," *doctoral thesis*, University of Hannover, 2002

[2] SOCRADES: Service-Oriented Cross-layer infRAstructure for Distributed smart Embedded devices, EU project EU FP6 IST-5-034116, <http://www.socrades.eu>

[3] K. Lorentz, „Ein Beitrag zur verteilten und adaptiven Materialflusststeuerung für Stückguttransport," *doctoral thesis*, Otto-von-Guericke-University Magdeburg, 2003

[4] H. Wurmus, B. Wagner, "Verteilte Steuerung in Java mit Interbus-Anbindung," in *VDI-Berichte 1515 „Industrielle Automation und Internet/Intranettechnologien"*, VDI Verlag, Duesseldorf, 1999, pp. 101 – 110

[5] R. Simon, M. Haenel, M. Riedl, "Verteilte Systeme durch Funktionsproxies," *Conference EKA 2010*, Magdeburg, May 25 – 27, 2010, Proceedings

[6] E. Newcomer, G. Lomow, *Understanding SOA with Web Services*. Addison Wesley, 2005

[7] N. Kakanakov, M. Shopov, G. Spasov, " Distributed Automation System based on Java and Web Services," *CompSysTech'06*, June 15 – 16, 2006, Tarnovo, Bulgaria.

[8] J. Seppälä, M. Salmenperä, "Intelligent Visualisation of Process State using Service Oriented Architecture," *IFAC World Congress 2005*, July 3 - 8, Prag, Proceedings

[9] Xiaohua Tian et al, "Service Oriented Architecture (SOA) for Integration of Field Bus Systems," in http://www.ece.iit.edu/~yucheng/YCheng_GC07_1.pdf

[10] D. Cearley, N. Gall, "Web-Oriented Architecture: The Path to Web 2.0," *Gartner Symposium Itxpro*, San Francisco, USA, May 14 – 18, 2006

[11] M. Wollschlaeger et al, "A Common Model for XML Descriptions in Automation," *IFAC World Congress 2005*, 04.-08.07.2005, Prag (CZ), Proceedings

[12] M. Wollschlaeger, Th. Bangemann, "Requirements, Structures and Content Model for Web-Based Maintenance Portals in Automation," *INCOM 2004 11th IFAC Symposium on Information Control Problems in Manufacturing*, Salvador (Brasilien), April 5 -7, 2004, Proceedings

[13] P. Knabe "Java-Applet als Runtime-Umgebung für eine Webprogrammierbare Steuerung (WPS)," *Report of the project work*, Duesseldorf University of Applied Sciences, Nov.2006

[14] R. Langmann, "Web-based Remote Control by LiveConnect," *International Journal of Online Engineering*, Villach, Vol. 1, No. 1, Oct. 2005, <http://www.ijoe.org>

[15] WebIO module: <http://www.wut.de/e-57630-ww-daus-000.php>

[16] Model factory of the Pi-LAB: <http://www.fabrik-21.de>

AUTHORS

R. Langmann is with the Duesseldorf University of Applied Sciences, Department of Electrical Engineering, Process Informatics Lab (Pi-LAB), Josef-Gockeln-Strasse 9, D-40474 Duesseldorf, (e-mail: reinhard.langmann@fh-duesseldorf.de)

Submitted June 13th, 2010. Published as resubmitted by the author July 13th, 2010.