LEAN WEB AUTOMATION

A new approach for automation of distributed systems

Prof. Dr.-Ing. Reinhard Langmann

University of Applied Sciences Duesseldorf, Process Informatics Lab (Pi-Lab), Josef-Gockeln-Str. 9, D-40474 Duesseldorf, Tel. +49-211-4351308, Fax. +49-211-4351308, R.Langmann@t-online.de, http://www.teleautomation.de

Abstract

The concept Lean Web Automation (LWA) describes a new approach for web-based operation of automation devices in distributed systems, which can be used flexible by means of compact and low-cost software without installation expense on the client side. A dynamic process data transfer in the TCP/IP network is implemented by a java-based application model. This model uses a process data proxy to create at web clients an interface to process data of a remote automation device. A first practical realization of the LWA is the software toolkit Web Access Kit for OPC, which uses OPC as the interface to the process.

1. Initial position

Since approx. 2 - 3 years, under the keyword "Internet technology for automation", the standard techniques from the world of the information technology are made available and also increasingly used for the automation of technical processes. This covers in particular:

- the application of Ethernet and TCP/IP as communication structure between elements of automation systems and
- the use of the Webbrowsers as universal control interface for automated devices, machines and plants.

Several well known research & development (R&D) works concerns itself e.g. with the extension and/or adaptation of the TCP/IP transmission protocol, in order to be able to use Ethernet systems also for real time communication (as a replacement for classical field bus systems). First results are present and be continued to advance by the market leaders in the automatic control engineering and several industrial alliances [1].

Foundation of this work are Ethernet and/or ethernet-based local area networks, in which functions of automation can be distributed free, but sometimes proprietary protocols are used. Shifting of these functions into the Internet is not intended in this work. By the using of Internet technology (Webserver, HTML, XML...) selected functions for limited tasks are realizable e.g. for the remote service over the Internet.

Other works integrates webservers in automation components (embedded Webservers) and creates thereby the possibility of an uniform access to automation plants by means of a webbrowser. These solutions are used either as web-based extensions to available process visualization systems (SCADA systems) or for the operating/observation and remote service purposes for embedded web-controllers over the Internet.

Web-based SCADA systems are usually very complex and expensive systems. Embedded web-controllers with browser-based operation possess only a small functionality. This is caused by the storage of the necessary operating systems as HTML pages inside the controller itself and here the memory-capacity is very limited [2].

Fig. 1 illustrates the state of the art on the basis of different structures with the Internet application for an automation device.

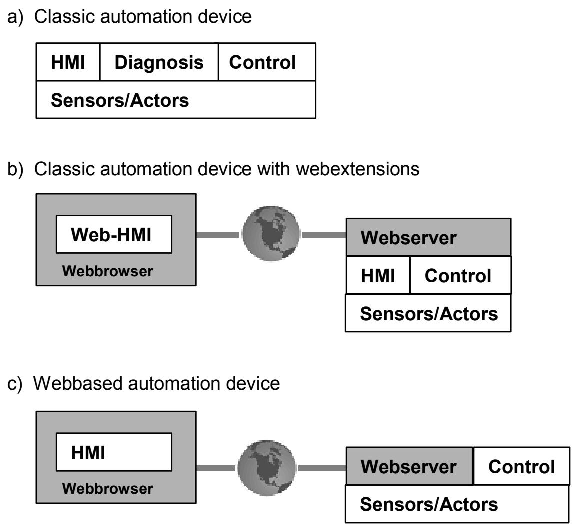

Fig. 1: Different structures of the state of art with the Internet-application for an automation device.

The goal of the known R&D work is a substantial improvement of the horizontal/vertical communication and data integration in automation systems, the reduction of interface problems by standardization, at last followed by the increase of flexibility and efficiency. The well known works assume that the basic automation functions (signal processing, control functions) still remain on a local automation equipment and be realized with a controller (PLC or PC controller, embedded Controller). Thus no changes in principle in the structure of the automation systems result.

First publications show in fact that the industry thinks about a completely different automation model e.g. the use of automation functions by means of "pay per use" [3].

2. Principle of the Lean Web Automation

The following contribution describes a solution for distributed automation on the basis of standardized Internet technologies, with which the automation devices attached in a net (Intranet/Internet) only deliver their process data (sensors/actuators) into the net.
The actual automation functions (operating, observing, processing, controlling...) are downloaded and used flexibly from resources existing in the net. Any computer in the net can take over the task of the system master for a automation system. This solution is called in the following *Lean Web Automation (LWA)*.

Task-adapted flexible solutions

With the LWA the production of task-adapted and flexible solutions is possible fast and cost saving. It applies the principle: For simple tasks also simple solutions are used. Only with rising task complexity the solution expenditure and thus the costs increase. This means e.g. for the visualization of only a few process data in the Intranet for many users that a complex process visualization system is not necessary, because of only one web page is loaded and implemented with the adapted visualization function as resources on the system master.

Process data in the net

Principal item of the LWA is the fast, reliable and safe supply of the sensor and actuator data of automation equipment in the Intranet/Internet. This process data communication is realized by a javabased application model *W2<process>Proxy*. This model makes an interface available, in the kind of a proxy of the respective automation function in the process area of a webclient (browser), to the process data of an external automation equipment (Fig. 2).

The substitute symbol <process> is placed in the application model for the respective practical realizations of the appropriate process data interface, e.g. to a OPC server, a driver for a field bus system or an embedded controller.

The W2<process>Proxy is equal to a distributed application following the client/server architecture. The W2<process>Proxy server is an independent application, which is implemented on the process data server. The functions of the W2<Process>Proxy client are realized by an Java applet. The applet is stored in the memory of the process data server (s. Fig.2).

In case of the request of a process data client the applet program code is downloaded by the process data server via its webserver and proceeded in the webbrowser of the process data client. The process data client (any computer in the net) takes over the function of the master for the automation device.

The W2<process>Proxy client creates a permanent dynamic data link to the W2<process>Proxy server in the process data server and, over the connection to the process data interface, to the actual process data of the automation device. The data connection between the W2<process>Proxy client and server is reached with the help of the Java-technology available for the realization of dynamic communication in distributed applications, Sokket API or RMI (RMI = Remote Method Invocation). The processdata exchange is made by an own OPC-like telegram protocol.

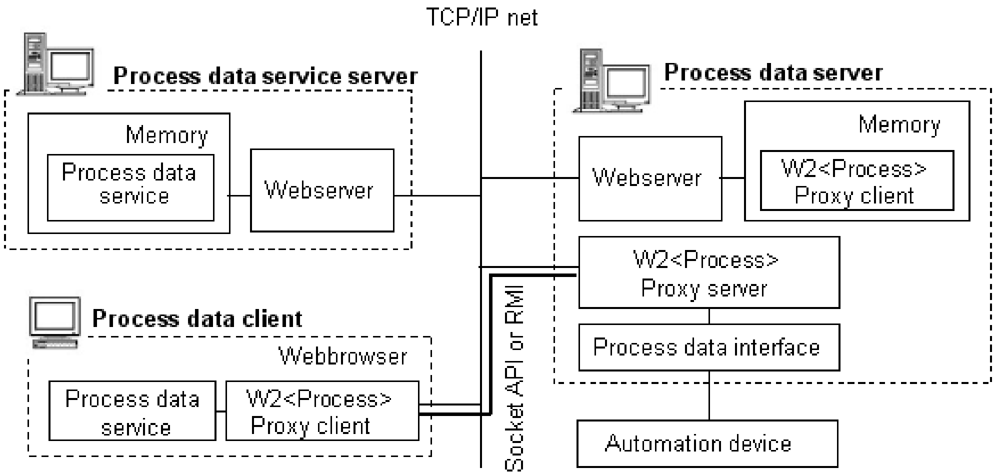

Fig. 2: Distributed arrangement for the Web-based operation of automation devices by the LWA concept

Distribution of automation functions

The processing (e.g. a graphic visualization of processdata) is made by process data services, which are downloaded from a process data service server via the

Web on the process data client and are also executed there.

Under a *process data service* in the LWA the program code is designated, which is necessary for the execution of the automation functions. A process data service can contain operation and observation functions and/or functions for the processing of process data in the sense of the processing of a control algorithm. So it is possible to visualize e.g. current trends of analogous process data (temperature, pressure, speed,…) in real time by means of a suitable process data service (e.g. a graphical presentation in a coordinate system) in the processdata client.

The program code of a process data service is e.g. in a ECMA Script conformal programming language (e.g. Javascript, JScript) created macro, which can be organized and treated comfortably as a separate *.js file. In addition, a Java applet or other Webobjects (ActiveX object, XML object) can be used as user specific process data services.

The communication between the W2<process>Proxy client and the process data service takes place on requirement as well as event-controlled over a special programming interface (script API).

Secure process data access

For a secure and reliable industrial application, additional methods on the level of the process data communication are embedded in the LWA concept beside the well known Web technology. Associated are e.g.:

- *User administration:* For the access to process data a flexible and fine adjustable User administration with group rights is integrated.
- *Access and error logging:* All accesses on the process data are logged related on user, process date and webclient.
- *Error management in case of client-crash or connection abort:* A connection interruption is recogniced in the process data server by the W2<process>Proxy and can be evaluated as a function of the application (e.g. all outputs to zero, creation of a new connection).

3. Realization with the Web Access Kit

The Software tool *Web Access Kit for OPC* is a first practical realization of the LWA concept, which uses OPC as an interface to the process [4].

The access to the OPC interface is realized by an own JNI-based library *J2OPC* (JNI – Java Native Interface). The data connection between client and server of the distributed application uses the socket-API.

The W2OPCProxy applet is embedded in a webpage by an HTML editor. Designations of necessary OPC variables (Item-IDs) of a certain OPC server are indicated in the parameterset of the applet including other configuration attitudes. The parameterset can be created manually by the user or be generated automatically from a user-friendly WAK wizard.

The script programming interface (API) of the W2OPCProxy applet is developed according the OPC interface and makes methods available for read/write access to one or more process variables.

The variables and/or items are addressed over their client handles. At the number of [n] items in an applet the individual items are assigned to client handles from 0 to [n-1]. Either the appropriate client handles or the values which will be written are handed over to the methods for the access to several items. These are then, separated with blanks, packed into a string-variable. The results of these methods are returned with data objects by the type `ReadResult` and `WriteResult`.

An event-controlled notification of a process data service by the W2OPC-Proxy applet as consequence of value changes of a process date is made by the triggering of a special function `onDataChange`.

The code example in Fig. 3 shows a use of the script API of a W2OPCProxy applet. A requirement based read access takes place via the call of the function `read-`

```
<script language="JavaScript">
<!--
function readFiveItems() {
  var i;
  var clientHandles = new Array(0, 1, 2, 3, 4);
  var strClientHandles = clientHandles.join(" ");
  var readResult = document.OPCProxy.readItems(clientHandles.length,
                                               strClientHandles);
  if (readResult.result == 0) {    // Wenn das Lesen erfolgreich war
    var formIndex;
    var strValues = "" + readResult.strValues;
    var values = strValues.split(" ");

    for (i = 0; i < readResult.numItems; i++) {
      formIndex = clientHandles[i];
      document.itemValuesForm.elements[formIndex].value = values[i];
    }
  } else {
    alert("Das Lesen war nicht erfolgreich!");
  }
}

function onDataChange(dchEvent) {
  var i, formIndex;
  var strClientHandles = "" + dchEvent.strClientHandles;
  var strValues = "" + dchEvent.strValues;
  var clientHandles = strClientHandles.split(" ");
  var values = strValues.split(" ");

  for (i = 0; i < dchEvent.numItems; i++) {
    formIndex = clientHandles[i];
    document.itemValuesForm.elements[formIndex].value = values[i];
  }
}
//-->
</script>
```

Fig. 3: Code example from the programming interface of the W2OPCProxy with JavaScript

FiveItems. For event controlled notification with changes of value the example implements the function onDataChange. In the example the read and/or with a data-change event received values of variables are shown in text fields of a HTML formular with the name itemValuesForm.

The WAK tool is also already available for several embedded Webcontroller. So e.g. for the ethernet based I/O module “Web-IO” of Wiesemann&Theis (Wuppertal, Germany) or the “BC660” of Beck IPC (Wetzlar, Germany), a very compact controller with embedded webserver and ethernet interface.

The *WAK for OPC* as well as the *WAK for BC660* also contain beside the W2<process>Proxy a few parametric changable visualization applets for the fast generating of simple user interfaces under use of digital/analogous process data (example Fig. 4).

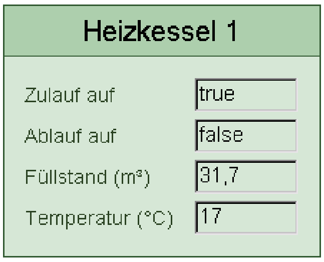

Fig. 4: WAK visualization applet for reading of analogous processdata of a remote controlled boiler.

4. Application

With the Lean Web Automation many time critical user activities at remote controlled machines and plants can be realized with standardized Internet technologies apart from common operation and observation tasks. Additionally, data acquisition with time-near data analysis and control tasks are also possible in the non time-critical area (temperature control, clima control, lights control...).

In the following, some successful practical examples are presented.

Operating and observing

Emphasis for the industrial application of the LWA is at present the realization of browser-based interactive graphical user interfaces. Thus e.g. the W2OPCProxy is used at Saint Gobain SEKURIT (Stolberg, Germany, Carglass Manufacturer), in order to make a limited number of production data accessible for many users in the Intranet.

Data acquisition and processing

A consistent employment of the LWA principle with the use of process data services of a process data service server is tested at present with EDAG (Fulda, Germany, Subcontractor company for car manufacturing equipment). In this example during the startup phase of a passenger car manufacturing plant the process data are locally noted by means of a mobile process data server (realized on a notebook) and transferred into a data base accessible over the Intranet/Internet for storing and evaluation of the data.

Controlling

For the testing of the use of LWA controlling services two applications were developed:

Telepresence laboratory for the PLC training

For a telepresence practical course to the PLC training (conformal to the IEC 61131 standard) of automation engineers was developed a didactic automation plant. Not only dynamic process visualization and operation are accomplished in the webclient (system master), also the processing of control algorithms for the automation station which is connected over the Internet [5]. Fig. 5 illustrates the LWA component structure, on which the telepresence lab is based.

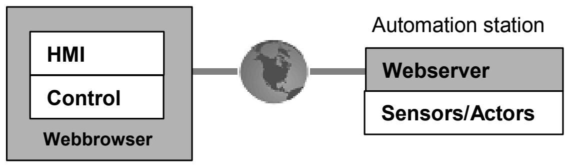

Fig. 5: LWA component structure for the telepresence lab

Fig. 6 shows a hardcopy of the realized Web-based PLC (WPC). The processing of the control algorithms is made in the browser by a process data service, which is realized as a Java applet with the function of an IEC 61131 editor and runtime environment. The system can be tested in the “Duesseldorfer Telelabor” under the URL: **http://www.telelabor.de**.

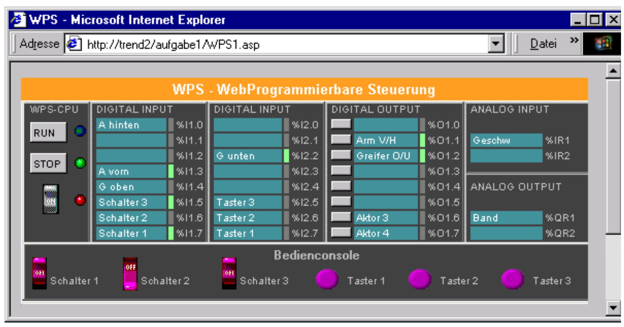

Fig. 6: Web-based PLC (WPC) based on the LWA concept

Traffic Light Control

For the demonstration of the efficiency of the LWA concept the control for a traffic light crossing using an ethernet controller with 16 digital in/outputs each (Web-IO) was built up. Basis forms a training course suit-case of the company Wiesemann&Theis (Wuppertal, Germany) which contains a traffic light crossing in small format. Observation of the traffic light crossing, the controlling and parametration are made by process data services, which are implemented in the webbrowser (Fig. 7). The control program is realized as a Javascript service (with approx. 300 lines program code) and is downloaded into the browser by a process data service server.

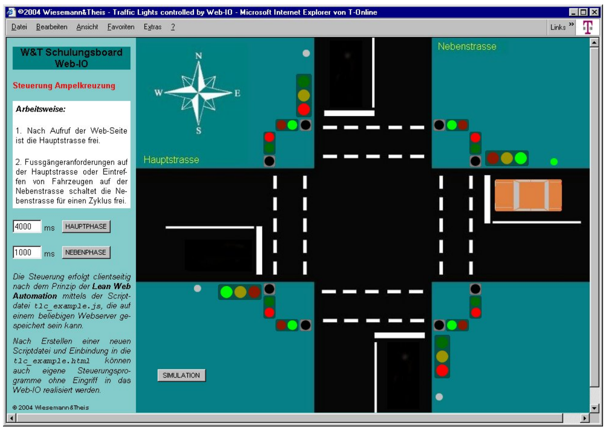

Fig. 7: Traffic light crossing with Ethernet I/O controller Web-IO

With the LWA solution an efficient and graphically responding use of the traffic light crossing control is possible despite the very much limited storage capacity of the Web-IO.

5. Results and operational experiences

LWA makes the efficient and flexible realization of Web-based automation solutions possible with the following characteristics:

- Process data in a Web-based process data client can be presented in two ways, requirement or event-controlled, with all abilities of a HTML programming (HTML, script languages, Flash, Java applet, VRML models, SVG graphics etc.).
- Apart from the realization of operating and observation functions the process data client makes possible also an application specific data processing e.g. for the controlling of the automation devices attached over the net.
- Process data services can also be downloaded and used by other server computers (process data service servers) attached at the network, which do not stand in connection with automation devices.

Apart from the unrestricted creative liberty of a Web-based HMI, the independence of the process data services from the process data server belongs to the advantages of the LWA concept. If many process data servers and/or automation devices of the same type are attached at the network, thus maintenance and a change of the respective process data services can be accomplished very simply and efficiently at only one place in the process data service server.

Several necessary process data services can be downloaded also from different process data service servers into the processdata client and executed there. One advantage is, that different and e.g. high-specialized process data services can be developed and maintained by different offerers.

The TCP/IP socket connection in the LWA concept makes low response times possible in comparison with other transmission techniques (e.g. SOAP, XML).

The employment of the W2OPCProxy with usual web-browsers (IE, NE, Opera) shows the following results:

- The Web2OPCProxy is usable also with data transmission rates in the Internet of < 10 kBit/s with non time-critical tasks of automation.
- With the arrangement in Fig. 2 could be obtained update rates < 100 ms for the process data refresh in a simple sequence control program (see Fig. 5) on the process data client over the Internet.
- A test operation of the W2OPCProxy in the Netscape Navigator (V4.78) with 10 process variables, whose values were changed at the same time, resulted in that the browser needs in average 7 ms starting from the moment of the receipt of a data-change telegram up to the illustration of the received values of variables in text fields of a HTML page. For the Internet Explorer (V5.5) the value amounts 32 ms.

6. Summary and future works

The flexibility and transparency of the Lean Web Automation correspond to a considerable degree to the future requirements at individuality and openness of automation systems during a simultaneous cost reduction in operation and maintenance. The first realizations of the concept are already used successfully in some projects. The further development of the LWA concept is continued in the following directions:

- Increased reliability of the process data access also with short connecting interruption and time delays.
- Investigation of the use and security problems with consideration of the passing of Intranet/Internet borders, coding of process data and resumption of aborted automation functions with automatic restart.
- Development of different services for the process-data processing and preparation in the Webclient.
- Testing of different visualization techniques for the LWA concept (e.g. with SVG graphics, flash animations and dynamic 3D VRML visualizations).
- Investigation of different service structures and operator models for the use of automation functions as services in the Web.

7. References

[1] *ZVEI:* Ethernet in der Automation. – Studie, 2003

[2] *Kühnel, C.:* Embedded Webserver mit Single-Chip Controller. - Script Verlag Kühnel, 2001, Wollerau

[3] *Kuehnel, C.:* Pay per Use. - Computer & Automation, 12/2002, pp. 36 – 41
[4] *Timerbaev, A.; Langmann, R.:* Webzugriff auf Prozessdaten. Teil III: Einsatz von Java. - *SPS-Magazin, 6/2002,Marburg,* pp. 43 – 45
[5] *Langmann, R.:* The German Remote Lab Network CONTROL-NET. – Proc. of 21th World Conference on Open Learning and Distance Education, Hongkong, 18 – 21 February, 2004, pp. 135 – 139

Veröffentlicht in:

Automation Panorama, 2004.08, No. 4, Beijing, 2004, S. 23 – 39