Process Control and Supervision through Internet

Authors:
Luis Payá, José M. Azorín, Nicolás M. García, José M. Sabater, Carlos Pérez, Ramón P. Ñeco
Departamento de Ingeniería de Sistemas Industriales, Universidad Miguel Hernández, Elche, Alicante, Spain,
ramon.neco@umh.es

Abstract — In this paper the control and supervision through Internet of a process managed by PLCs (Programmable Logic Controller) is presented. The proposed software architecture allows remote users to know the sensors state and to send control signals to process actuators, by means of a graphical interface that is accessible via a web browser. Also the remote user can download control programs to PCL. It has been used the client/server architecture, where the PLC and the server PC are connected by a Visual Basic application, and the server PC and the client PC exchange data using the TCP/IP protocol. This architecture has been applied to a real system that consists in an electropneumatic manipulator. This manipulator together the PLC and the server PC form the virtual laboratory. This kind of remote control has a shiny future in the industry and the education world. From an educational view point, it allows students to verify its control algorithms in a real system from its house, analyzing the process evolution using only the web browser, without any software else.

Index Terms — Programmable Logic Controller, process control, Internet, client/server model, virtual laboratory.

INTRODUCTION

The fast evolution of Internet in last years has produced that the Internet based systems play a very important role nowadays and in the future its importance increases bigger. In this framework, it appears the possibility of combining the process control and automation techniques joint the advantages that offer Internet as global network.

The purpose of this study is developing a base tool that allows the remote control through Internet of a process managed by PLCs [1]. The objective is to achieve that a student can access to automation laboratory equipment remotely, from his house, and he can test its control algorithms observing the process evolution.

This remote control can contribute new work and learning concepts, because of laboratory practice are a fundamental part in the engineering formation. The not physic presence of the student in the laboratory develops the individual learning as new formation strategy. This fact facilitate a more free learning, so the student must not follow the rhythm imposed in a practice class where there are dozen of students. The new control also allows that the student can access anytime to the virtual laboratory. In this way, the student can analyze with a bigger emphasis and longer time the aspects that he considers more relevant.

By the other hand, these techniques facilitate the optimization of laboratory equipment, so it is possible to share resources: some students can work with the same equipment, and the students can use complex equipments because these equipments must not be physically present.

The developed tools can be extended also to the industrial world, allowing, for example, the following applications:

- Remote maintenance and reparation of equipments and installations. In this way, the machine manufacturers can offer a maintenance service online of its machines.
- Interaction with processes in companies that have branches distributed around the world. The same operator can manage all the processes in a centralized way.

Nowadays there are some commercial tools that allow this kind of remote control. However, these tools have some inconvenient that have been overcome with our tools developed. The fundamentals characteristics of the tools developed are:

- Oriented to small and median applications.
- Portability: The client user only needs a PC connected to Internet with a web browser, without any software else. Each client can access to the process from any PC, since any special software installation is required.
- Open system: The software expert can decide which process variables are supervised and controlled in the server and the remote client, being possible that these variables are different in each side.
Remote download to the PLC of control programs, without any hardware additional. Also, only a text editor is needed to make the PLC program.

This paper is organized as follows. Section 2 shows the software architecture developed. Next, in section 3, the information flows produced during the supervision and control process are analyzed. The experiment performed is described in section 4. Finally, the conclusions and future works are exposed.

SOFTWARE ARCHITECTURE

The software architecture has been developed considering the two main objectives that it is desired to reach. These objectives are the next:

- Achieving that a user could supervisor and control the process remotely through Internet using any web browser. Since it has been considered processes controlled by PLCs, this objective is equivalent to make accessible the PLC inputs and outputs to the remote user. The input knowledge (sensors state) allows the supervision, and the output access (actuators) allows the control.
- Making possible the download of new programs to PLCs. These programs can be created using any text editor, so the user must not install the specific PLC software (Siemens Step 7) [3]-[6].

Client/server architecture has been implemented to reach both objectives. This architecture will do possible that a remote user can visualize in a web browser a Java applet where the process variables are accessible. To achieve that the applet can be visualized in any web browser, it has been included in an HTML page that resides in a server with Windows NT 4.0 powered by the Microsoft Internet Information Server software. To access the PLC variables from the applet, a Visual Basic application that is located in the web server has been developed. This application and the PLC S7-314 IFM are directly communicated. The application and the applet will exchange the information through the TCP/IP protocol using the sockets model [5].

In figure 1 the complete developed architecture is shown. The fundamental characteristics of these applications are:

Server Application

Developed using the programming environment Visual Microsoft Basic 6.0 for two fundamental reasons:

- Simplicity.
- Siemens libraries: Simatic Computing [2] that offers OCX objects and functions that allow the access to inputs, outputs and internal variables of the PLC and the Command Interface, with functions that allow us to access to the functions of Step 7.

To be able to supervise the process, a graphic interface should be implemented. This program, it could be used individually as a local Scada if there is not control necessity through Internet.

On the other hand, to facilitate the communication with the client PC, the TCP/IP protocol will be used using Sockets [5]. To do this, the Winsock control of Visual Basic is used.

Client Application

The chosen language for the development of the application is Java [4], due to the following characteristic:

- Distributed: Java provides a collection of classes for its use in network applications. These functions allow to open sockets and to settle down and to accept connections with servers or remote clients, facilitating in this way the creation of distributed applications.
- Applets: Java can be used to create applets; programs that appear embedded in web pages with the capacity of executing complex actions, like for instance to establish network connections. They are not dependent of the navigator that the client uses and even of the operating system. Furthermore, the applets have a very well defined space in the web page where they are executed, not being able to consent to the client file system unless the client permits it.

To build the graphic interface, Swing components have been used. A characteristic of these components is that you can specify their aspect and behaviour from the program, not having the aspect and behaviour predefined by the native platform. When the remote user makes some action on one of these controls (for example, to press a button), an event is generated. Therefore, to respond to the client actions, it will be necessary to recognize the source of the event, and to work in consequence.
This graphic interface consists of two parts, one of them is the process monitoring, where the client can observe the value that the characteristic variables of the process are taking in every moment, and the other one is the control of the process, in this part the user can act on the remote PLC by pressing buttons in the application.

So that the applet has access to all the variables of the process, the control application that resides in the server communicates through the TCP/IP protocol with the applet, using sockets. The functions to make this, is supported by Java through the class "Socket".

When one of the variables of the process changes, it is received through the socket a sequence of variables (bytes). The first value is an identifier of the variable that has changed and the rest bytes will indicate the new value. On the other hand, when the client acts on a button of the applet to carry out an action on the process, a sequence of variables (bytes) ,that indicate the variable to modify and its new value, is sent.

To allow the visualization of the applet, it is necessary to install in all client machines the java plug-in of Sun, starting from the version 1.3, downloadable from the official web page of Sun.

Applications for controlling the automatic wake-up and stop of the server program:

So that a remote client can consent to the process, it is necessary that the server program is running exactly when a client try to connect to the web page, and that it continues being executed until the client concludes its tasks. To avoid that the server program is continually running, waiting for connections of the remote users, an application that take into account when a remote client request the process web page and when this client leave the web page must be implemented.

This can be carried out by an auxiliary Visual Basic application residing in the server and a second Java applet that is running in the same web page that the applet of the process. The sequence to program would be the following one:

- The auxiliary application will always be being run in the server, without any graphic interface neither direct access to the process and due to the characteristics of this application, very few resources of the system will be consumed. It will have an open socket waiting for new connections.
- When the client connects to the process web page, the auxiliary applet will establish contact with the Visual Basic application, sending it a message through the socket.
- When the message is received, the VB application will check if the server application is running or not.
  - If it’s running, it means that another client is already connected, of which the auxiliary applet is informed by a message through a socket. Then the communication will be cut and the VB application will stay listening for new connections.
  - Otherwise, the application will be opened up, using the functions of the API of Windows, the client will be informed by a message through the socket and the communication will be cut and the VB application will stay listening for new connections.

The closing of the server application can be carried out from the same server application, recognizing when the contact with the applet client is lost.

Applications for downloading programs to the PLC

As it has already been commented, the functionality of downloading programs to PLC can be implemented in the VB server application, using the functions provided by the Command Interface library from Siemens. Command Interface library is an automation interface based on OLE elements which allows us to use the native functions of Step 7. It is necessary to keep in mind that any Simatic project can be compressed in two text files: the source file, with " .awl " extension and the " .sdf " symbol table, that will be necessary if in the source file, some memory addresses of the PLC have been replaced by mnemonics to facilitate the programming task.

The applet of the process will have to contain a button “Download the program” that, when is pressed, produce an event that informs the server, by a message through a socket, that the client wants to download a program to the PLC. When the VB server program receives this message, the following sequence of operations will be carried out:

- Creation of an empty Simatic project.
- Import of the source file and symbol table to the new project.
- Compilation of the project.
- Downloading the blocks compiled to the PLC. For do this, the CPU must change its state to STOP.
- Put the CPU in RUN and run the user program.
- Capture of errors.
Obviously, so that this discharge can be effective, the files that contain the source code of the program should reside in the hard disk of the server, so it is necessary to carry out a transfer of files among the hard disk of the client to the server one. A possibility to transfer these files is the developing of a HTML form and a CGI application that receives the data of the form and processes them. A HTML form is a document section that contains elements called controls, such as buttons, menus, etc. The user usually completes the form modifying these controls and then sends the data to the server where will be processed. On the other hand, CGI (Common Gateway Interface) is a standard for the interaction of the external applications with web servers. In other words, a CGI application is an executable program that is executed in the server, and it will allow us, among other things, to give answer to the user actions on the HTML form. On this form, the remote user will specify the files to transfer and will press the button “Send”. Then, the files will be sent to the server and the CGI application developed will receive them by the standard entrance, its function is reading the data and keeping the files in the hard disk of the server.

**DLL library**

It was programmed by the programming environment: Microsoft Visual C++ 6.0. This library contains the necessary functions for closing the windows that appear when a project is downloading to the PLC from the application program. The use of these functions becomes indispensable when we load a program to the PLC, since in this situation certain windows in the server PC are opened up and must be closed so that the execution of the program continues. This window closing should be made by pressing a button of these windows. Since in the server PC there is not any person that can carry out this work, this task should be done by the server application in an automatically way and in a completely transparent way to the remote user. For do this, the functions of the Windows API are used. As from a Visual Basic application is impossible to make a click on a button of another window, it has been necessary to develop this library in C++.

**Applications of security**

To impede the free access to the process web page, the initial page that appears when a user wants to connect to it contains a HTML form in which the user should introduce his login and password to validate his access. These data are checked by a CGI application programmed in Visual Basic that searches in a database and returns as answer the web page that contains the applet of the process, if the user's data are correct.

**Images Server**

With the purpose that images of the production process can be visualized through Internet, an images server AXIS has been installed and a Sony EVI D30 camera has been connected to it. With this server, the pan, tilt and zoom of the camera can be controlled.

**FLOWS OF INFORMATION IN THE SYSTEM**

To clarify the operation of the system, the flows of information that is taken place when the client is monitoring and controlling the process, and when a user program is discharged to the PLC are exposed. These flows of information are shown in the figure 2.

**Supervision of the process**

1. When a process variable changes, a sensor ,that is monitoring the process, captures this event and sends the new value of the variable to the corresponding (digital or analogical)input.
2. The server program with a graphic interface will have an OCX object associated with this input. When a variable change is taking place, an event will be produced in this object and the new value will be read and the interface will be updated.
3. Then, The identification code of the variable and its new value will be written in the socket.
4. The client will receive these messages through its socket and then will identify the variable that has changed and update its own graphic interface.

**Control of the process**

To control the process, the same actions in different direction (from client to the server) are carried out by the applications developed (see figure 2).
Downloading programs to the PLC

In the first place, the project files should be transferred from the client to the server. This transfer is carried out using the TCP/IP protocol and an CGI application residing in the server. This application receives the data files and keeps them in a predetermined directory of the server hard disk. Then, the client gives the order of load the program transferred to the PLC.

EXPERIMENTS RESULTS

The goal of the project consists of the development of a tool that permits the supervision and control of any industrial process through internet. However, to demonstrate the validity of the obtained results, they have been applied to a scaled industrial process in the laboratory.

In figure 3 a graphically distribution of the elements used in the laboratory setup is shown. In figure 4, a real image of the experimental setup could be seen. On the other hand, in figure 5 a detail of the graphic interface of the server application is shown.

CONCLUSIONS

To finish the study carried out on the monitorization and control of a process through Internet, the basic characteristics of the project are recapitulated:

- The necessary functions to supervise and to control locally or remotely a process connected to a PLC have been implemented.
- A minimal installation is required on the part of the client: only a PC with access to Internet, a text editor, a navigator web and the plugin of Java, downloadable freely from the web page of Sun.
- The discharge of PLC programs from the remote PC to the PLC has been developed too.
- The capture and treatment of the most common errors has been implemented.
- The remote user has a double source of information: the supervision software and the real images of the process.
- Some minimum consistent requirements of security have been developed too.
- It is not necessary that the server application is running continually in the server PC.
- Everything has been developed to make the use of this software easy to the remote user.

REFERENCES

[2] "SIMATIC Computing User Manual 6ES7 673-6CC01-8BA0"
FIGURES AND TABLES

FIGURE 1
SOFTWARE ARCHITECTURE DEVELOPED

FIGURE 2
SYSTEM INFORMATION FLOWS
A) SUPERVISION AND CONTROL
B) DOWNLOADING OF PLC PROGRAMS

FIGURE 3
GRAFICALLY DISTRIBUTION OF THE LABORATORY SETUP

FIGURE 4
IMAGE OF THE ELEMENTS USED IN THE EXPERIMENTAL SETUP

FIGURE 5
GRAPHIC INTERFACE – LOCAL SCADA IN THE SERVER PC AND WEB PAGE DEVELOPED