

Web Development Techniques and Remote Laboratories

[doi:10.3991/ijoe.v5s1.906](https://doi.org/10.3991/ijoe.v5s1.906)

D. Ursutiu, D. Iordache, P.A. Cotfas, D.T. Cotfas and C. Samoila
Transylvania University of Brasov, Brasov, Romania

Abstract—In the actual context of the world economic and financial crisis, any source that causes economy is welcomed. In the technical educational system such a source is represented by the implementation of remote labs. These allow, among other things, the sharing of modern technology between several centers and also the saving of power by using an adequate energetic management. This paper presents a solution for implementing remote labs by using web technologies that are platform independent from the client point of view. Such a lab implementation exemplification is materialized by remote controlling the NI-ELVIS platform.

Index Terms—Remote Laboratories, Ajax, MySQL Database, LabVIEW, NI-ELVIS.

I. INTRODUCTION

The rapid growth of the Internet and its subsequent user base have made it a powerful tool for obtaining or sharing knowledge at a much more rapid pace than ever before. These aspects allowed the appearance of many online remote labs or e-learning platforms set up by different universities or associations from all over the world.

The motivations behind setting up such a system may vary from the wish to grant access to information or hardware to others, to the reducing of the overall cost of deploying such systems. These systems have known great success in universities and have proven themselves again and again. But what about the industry point of view?

Such a control and visualization system might come in handy in an industrial environment where there might be a need to monitor the state of the machines working on the factory floor. The system can monitor the performance, speed, status or yield of each machine and display this data to a factory executive in a timely and elegant manner, over the local Intranet or even the Internet, eliminating the need for actual presence on the factory floor.

During the last few years a number of techniques have been implemented in order to allow the control and monitoring of remote devices. A number of different remote control interfaces have been developed on platforms such as Java, Flash or some other virtual machines, many of which may be proprietary and thusly affect the cost of implementing a remote engineering system.

However, by far, the most commonly used technology is the Java Applet, which has been used successfully in such a situation for a few years now, but with a few setbacks that affect some of the other technologies mentioned. The software still needs to be installed on the machine. This might be problematic on networks where the administrator prohibited the installation of additional software. Also, Java and Flash tend to take a long time to

load, and installation might be complicated on some machines, this goes especially when working on alternative platforms or operating systems.

II. OBJECTIVE

The objective of this paper and of the project detailed in it is to eliminate some of the problems presented earlier and to offer solutions for implementing a remote lab independent on the operating system – no need to install anything – and even platform independent, from the client point of view.

The main concept of this system is the client-server technology.

Because of the rapid development of “web 2.0” technologies, the current web design and web programming techniques provide more than enough material to work with the programming technique that is called Ajax, which stands for Asynchronous JavaScript and XML. The client Ajax interface is the part that will run in the web browser (completely platform independent).

On the server side, the graphical programming language LabVIEW is used. This is used for creating control and measurement virtual instrumentation dedicated to the accomplishment of laboratory works. The LabVIEW applications will be sending and receiving data from the browser, communication between the two will be established through a MySQL database. MySQL is perfect for web applications, being free, easy to use, fast and also safe.

Ajax is not actually a programming language; it’s just old languages gathered together and performing together to form a new programming technique. The most important aspect is that the web page will not be refreshing itself, all the server requests are executed in the background, the user is not aware of them.

The system basic operating principle is that LabVIEW will be writing the data in a MySQL database, from where the website will extract that data and present it to the user, the speed of the whole process is highly dependant on the network speed. The process can also be reversed: the website will be sending data and LabVIEW will be reading it from the MySQL database.

The below pictures illustrate the web client and LabVIEW interfaces for a simple application. This example shows how the data can be sent between client and server via network for controlling a DC power supply.

III. TECHNOLOGIES USED

As there are many software solutions coming together in this application a more detailed description may be re-

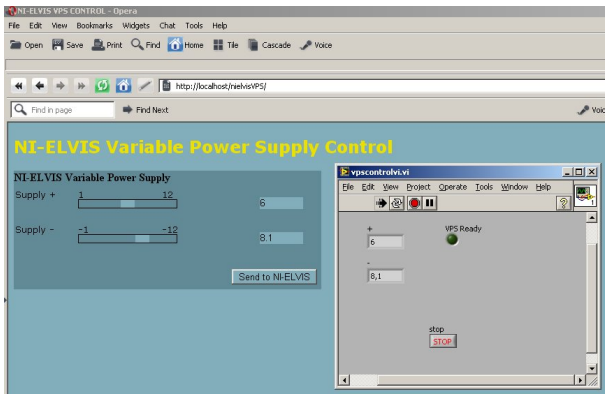


Figure 1. Controlling a DC power supply from the web application

quired. These can be classified in two categories, software running on the server system and software running on the client system.

The server applications include LabVIEW, Apache HTTP server (with the php component installed) and the MySQL database.

LabVIEW is a graphical programming environment developed by National Instruments

This environment allows the easy designing of virtual instruments for real measurements and analysis. The interfaces designed in LabVIEW are friendly and easy to understand, being very similar to the real instrumentation. LabVIEW also allows the interconnecting with data bases and with server type applications.

The Apache Web Server is the software used to send the website files and data to the client. Any HTTP web server may be used instead but Apache is one of the most popular one used on the web. It is developed and maintained by the Apache Software Foundation, an open community of software developers, and freely available for commercial use under a free custom license (Apache License 2.0).

MySQL is a multithreaded, multi-user SQL database management system initially developed by MySQL AB and currently owned and maintained by Sun Microsystems. Similar to the Apache solution, many other databases may be used but, again the MySQL database is the most popular database solution for websites with reportedly over 11 million installations worldwide. The license under which it is released is the GNU General Public License, making it free for use. This aspect of free software and the many other characteristics of MySQL make it a perfect choice for the application at hand.

While MySQL follows the standard SQL syntax it has some parts where they do differ. Besides these differences it has some distinguishing features apart from other databases such as the possibility to choose from many storage engines (native, partner, community or custom) which allows a great flexibility.

The application also requires a server-side scripting language, a role played in this case by PHP. PHP is a recursive acronym standing for PHP: Hypertext Preprocessor. It is currently developed by the PHP group. The syntax used is similar to the C syntax, with other influences coming from Perl or JAVA.

PHP is released under the PHP License which is a non-GPL free open source license, evidently making it free for use and modify.

Even though it is not necessary to use PHP as a server-side scripting language, again it is well suited for creating websites, being the most popular Apache component used on the Internet up to date.

These three software server components come together to build a free, powerful, easy to use, easy to implement development and website serving environment, making developing and maintaining complex multiuser database, driven internet applications with virtually no limits.

The client application is basically a website, an Ajax application or rich internet application. As mentioned before Ajax stands for Asynchronous Javascript and Xml. This enumerates some of the technologies used to create the client-side part – the website. The basic element is the JavaScript scripting language.

JavaScript is an easy to use scripting language which loosely follows the Java and C syntax but its design principles are derived from the Self programming language. There are several JavaScript dialects but the most popular one is the one used for client side html scripting which is found in a great number of websites. Originally it was called LiveScript and bundled with the Netscape Navigator browser from Netscape (the developers) soon after the name changed to JavaScript and more browsers included the runtime engine. Currently all the major web browsers (Internet Explorer on Windows, Safari on Mac, Opera, all the Mozilla based browsers – OS independent) support JavaScript. This aspect, in turn assures that the application is OS and browser independent.

XML stands for eXtensible Markup Language and it is a specification for creating custom markup languages. Its basic use is to send data structured in a custom way over an information system such as the Internet. Thusly, a user may create custom “tags” for structuring and sending text data.

Along with these main technologies come the basic website building tools HTML and CSS. HTML (Hypertext Markup Language) is another markup language, similar to XML (both have been derived from SGML – standard generalized markup language) only much stricter in the sense that the name of the “tags” (also called elements) have been predefined by the developer. HTML is used to inform the browser about the way the text data is structured in the website. Recently a stricter more standard way to structure a website has been released – XHTML (eXtensible Markup Language), however this is still HTML with a few new rules. HTML or XHTML are NOT tools to create the website design or layout. This is where CSS comes in.

CSS stands for Cascading Style Sheets and is a set of rules created to “style” the HTML elements. Properties like positioning, color, font face and many more can be added to the HTML page to create the layout. There are three types of style rules that may be applied: external CSS file (independent from the HTML file), CSS style set in the header section of the HTML file and inline styles which are applied directly to the HTML element. The “cascading” property of CSS tells the browser to apply rules in the external file first, then the rules in the header and finally the rules in the inline style. This means that if a certain rule is applied to an element in the external file it is overridden by a rule in the header section and overridden again if the inline style applies more rules. In conclusion, the inline style has the topmost priority to apply rules to

an element. HTML and CSS are the basic building blocks of a simple website.

XML, (X)HTML and CSS are all developed and maintained by the W3C (World Wide Web Consortium) and have been standardized.

IV. SAMPLE APPLICATION

The following paragraphs describe the usage of Remote Lab in Ajax technology: Controlling the ELVIS (Educational Laboratory Virtual Instrumentation Suite) workstation developed by National Instruments.

The application is currently under development for the second version of existence. Shown below is an image with its predecessor, a proof-of-concept application [1]. The client interface is running on Internet Explorer 6 (right), and the server application (left) is running on LabVIEW 8.2.

The user was able to control the NI-ELVIS Function Generator and the Variable Power Supply, and also get data from the Digital Multimeter and Oscilloscope.

The current version has an improved interface and additional features; it will allow control of the power supply, the function generator and the digital outputs. Also, it will monitor the digital multimeter and the oscilloscope.

The basic operating principle of the application is as follows: the DAQ board performs a measurement (from the NI-ELVIS station) then sends the data to LabVIEW.

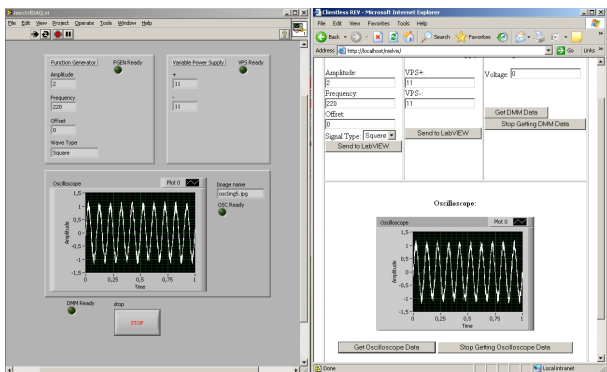


Figure 2. Version alpha 1.0 of the NI-ELVIS control web application



Figure 3. The goal of this version was to include linear potentiometers in an effort to make the interfaces as similar to an actual device as possible

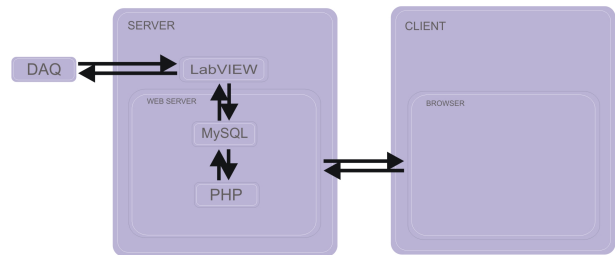


Figure 4. Graphic of the basic operating principle

After formatting or operating on the received data LabVIEW writes it to the database, and then flags it as ready for the client. At this point, most of the server's job is done, because the PHP call is initiated from the client interface.

The PHP script reads the values from the database, and then returns it through XML to the client interface running in the website.

For sending data back to the server the process is reversed, as detailed below.

REFERENCES

- [1] D. Ursutiu, D. Iordache, P.A. Cotfas, D.T. Cotfas and C. Samoila "New Technology Used in Remote Laboratories" Remote Engineering and Virtual Instrumentation REV2008, Dusseldorf – Germania, June 23-25, ISBN 978-3-89958-352-6, 2008;
- [2] Cotfas P., Ursutiu D., Samoila C., "Self Growing Remote Controlled Laboratory", International Journal of Online Engineering i-JOE, Vol 2, nr 1, ISSN: 1861-2121, 2006;
- [3] Luis Gomes, Javier Garcia-Zubia, "Advances on remote laboratories and e-learning experiences", University of Deustos - Spain, 310pag, 2007, ISBN 978-84-9830-077-2;
- [4] <http://www.ajaxian.com/>
- [5] <http://www.w3.org/Style/CSS/>
- [6] <http://www.w3.org/XML/>
- [7] <http://www.apache.org/>
- [8] <http://www.mysql.com/>
- [9] <http://www.php.net/>
- [10] <http://www.ni.com/labview/>

AUTHORS

D. Ursutiu, Prof. Dr., Transylvania University of Brasov, Department of Physics, 29 Bd-ul Eroilor Brasov, cod 500036, Romania, uodoru@unitbv.ro

D. Iordache, Student, Transylvania University of Brasov, Department of Physics, Engineering Physics Specialization, 29 Bd-ul Eroilor Brasov, cod 500036, Romania, dragosh.iordache@yahoo.com

P.A. Cotfas, Lecturer Dr., Transylvania University of Brasov, Department of Physics, 29 Bd-ul Eroilor Brasov, cod 500036, Romania, pcotfas@unitbv.ro

D.T. Cotfas, Lecturer Dr., Transylvania University of Brasov, Department of Physics, 29 Bd-ul Eroilor Brasov, cod 500036, Romania, dtcotfas@unitbv.ro

C. Samoila, Prof. Eng. Dr., Transylvania University of Brasov, Department of Science of Materials, 29 Bd-ul Eroilor Brasov, cod 500036, Romania, csam@unitbv.ro

This article was modified from a presentation at the REV2009 conference in Bridgeport, CT, USA June 2009. Submitted, April, 21, 2009. Published as resubmitted by the authors on July, 20, 2009.