Labicom.net – Putting Online LabVIEW Based Remote Laboratory in Less than Four Minutes with WebPager Tool

http://dx.doi.org/10.3991/ijoe.v9iS8.3374

Igor Titov[1,2] and Evgeny Titov[2]
[1] Labicom.net – the online laboratories platform
[2] Bauman Moscow State Technical University (BMSTU), Moscow, Russia

***Abstract*—Most of currently existing laboratory servers of remote labs were built fully or partially with LabVIEW programming language from National Instruments. While proving to be the best tool for fast development of excellent SCADA and data acquisition applications, it appeared to be inadequate when it comes to deploying an application to the web. This paper will analyze the tools that can be used for accomplishing this task in the remote laboratories. Various National Instruments and third parties technologies will be compared. It will be shown that LabVIEW application serving as laboratory server can easily be deployed on the standard pluginless web-page without any modifications in LabVIEW code with WebPager.**

***Index Terms*—LabVIEW, laboratory server, WebPager, HTML5 web page, real-time user interaction.**

I. INTRODUCTION

Most of remote laboratory projects start from building the application controlling remote experiment. A natural choice for this kind of applications is LabVIEW. Next task after controlling application is ready is to put it online for remote access by students.

LabVIEW applications can't work outside LabVIEW Run-Time engine of a particular version that is needed to be installed on each remote client machine. Moreover, to access such application from browser a special plugin from NI must be loaded. This technology is called LabVIEW Remote Panels [1] and has a number of limitations. One of them is limited property nodes functionality. Another is that it consumes inappropriate amount of web traffic.

National Instruments recognized implications of Remote Panels for developers and suggested more cross-browser friendly solution – NI LabVIEW Web UI Builder based on Microsoft Silverlight technology [2]. It allows LabVIEW developers to create browser applications with simplified LabVIEW. Interfaces created with this technology are smooth looking and modern. But it also appeared not to be convenient enough requiring Silverlight plugin on remote clients. It also usually requires opening custom ports and not truly cross-platform.

Another most recent attempt from National Instruments is Data Dashboard. This application is also able to connect LabVIEW application to remote device but functionality varies across platforms, user interfaces need to be developed and they are quite limited. It can be programmed only from the tablet, requires changes in the code, needs opening ports and poses difficulties for using encrypted communication channel.

More elegant solution for remote labs is NI Web Services. It produces cross-platform output and based on standard Internet technologies. But Web Services may consume lots of traffic, create network bottlenecks and introduce delays because it utilizes RESTful API (based on HTTP). Also it requires great development effort to make remote lab working on Web Services.

There were external to NI attempts to build a bridge between a LabVIEW application and a web-browser. For example, researchers at EPFL used Java-based JIL (Java-Internet-LabVIEW) layer for this end [3]. However, they abandoned it in favor of pure HTML pages because Java Applets needed for JIL operation are becoming extinct in modern browsers. Yet another solution is LabSockets [4] providing access to LabVIEW application from a web-page. However it has a few drawbacks that convinced us to develop and use WebPager [5] tool presented in this paper.

LabSockets is the closest to WebPager solution because it also uses web sockets protocol as communication layer and automatically generates web-pages from LabVIEW front panels. LabSockets generates web-pages that only distantly resemble original front panels.

Functionality of LabVIEW controls and indicators is not taken into account in LabSockets, and there are many limitations prohibiting its use in complex graphical user interfaces (e.g., only four plots per graph are supported and they ignore all its options). There are many other limitations too. Also LabSockets is not optimized for remote laboratory tasks and not integrated into any remote laboratories framework.

During Labicom.net platform development we recognized the necessity to reuse existing LabVIEW code and to deploy remote laboratory clients on the web. WebPager tool was developed to put LabVIEW front panels into standard HTML5 enabled browser not requiring any plugins. However, it should be mentioned that this type of solution is not a remedy for all kinds of remote laboratories as it provides direct access to laboratory server for students. We suggest that for complex remote laboratories like Remote Laser Laboratory at BMSTU [6] it is much better to have a division of labor between client and server applications and not to try to mix both in one single application. For simple remote laboratories or laboratories using standard hardware platforms similar to NI ELVIS it

might be useful to take an advantage of WebPager. It allows laboratory administrators to have their laboratory up and running in the web-browser in less than five minutes which was shown during EXPAT'13 demonstration [7]. Being a product of Labicom.net [8], [9] it is fully integrated into this platform therefore automatically providing reservation system, load balancing, SSL-certificate and other necessary for remote lab features.

II. COMPARISON OF UNDERLYING TECHNOLOGIES

Client side application of the remote laboratory needs to be as accessible by end users as possible. Students should be able to access remote lab from any device and from any network. Thus, it should be cross-platform and require minimum to no changes to network and firewall configuration. All mentioned above technologies except for Data Dashboard work with web browsers.

It is guaranteed that only two ports are opened on client machine: 80 (HTTP) and 443 (HTTPS). It means that Data Dashboard, Remote Panels and UI Web Builder can't provide ubiquitous access to remote experiment because they require configuring ports and network settings. Web Services meet this requirement but they use plain HTTP(S) protocol (REST API).

Using HTTP(S) for bidirectional real-time communication is not efficient in terms of traffic and delays. Servers are not able to send information to clients via HTTP(S) without a request and HTTP(S) is a stateless protocol. Because of this clients need to constantly poll servers with HTTP(S) requests (POST method). Each HTTP(S) request and corresponding response may easily contain header information of more than 1000 bytes, and 0 bytes of payload data. It makes bidirectional real-time communication via plain HTTP(S) very inefficient. Even when there is data the ratio of payload to header is still very small. For example, for boolean variable (1 byte) and HTTP POST header of 1000 bytes it will be 0.001.

Various technics under umbrella name COMET are used for alleviating such inefficiency. These workarounds use long polling where client sends HTTP request but server doesn't respond and keeps connection open until it has needed data. In the end the connection is closed on timeout and client resends request again. It is possible to decrease server polling rate but the problem remains. Moreover, such solution is difficult to build and prone to errors. That is why LabVIEW Web Services may solve remote lab connectivity problems in some cases, but in a very inefficient way.

The problem of bidirectional real-time communication between server and client is solved by a new Internet protocol called web sockets [10]. It is able to work over the same ports as HTTP(S) and it doesn't have the overhead that POST requests have.

Thus, web sockets technology allows connecting remote laboratory to the browsers of end users in a very efficient way.

WebPager software described in this paper uses web sockets as underlying data communication mechanism.

III. WEBPAGER DESCRIPTION

WebPager puts LabVIEW front panel online without requiring any changes in the LabVIEW code. Although the API for advanced features could be used, in the simplest scenario one can put front panel online by choosing corresponding menu option in the *Tools >> WebPager* menu in LabVIEW.

WebPager will generate HTML5 code and upload it to the server providing real-time user interaction in both directions, i.e. any changes made by user in the browser will be reflected in the front panel and vice versa. Typical average delay is 300 milliseconds. If there are no changes in control or indicator values, then no data being sent.

WebPager will preserve most of the LabVIEW options for each GUI element (figure 1).

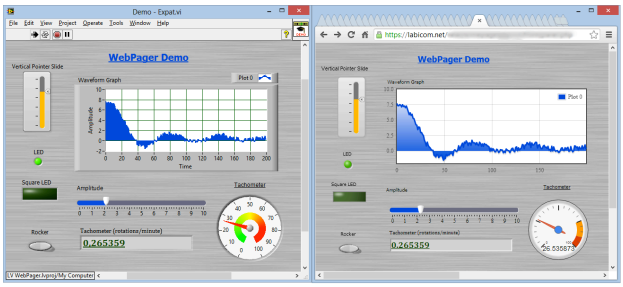

Figure 1. WebPager automatic web-page generation (LabVIEW front panel on the left, browser web-page on the right). Most LabVIEW controls and indicators are supported. Styles and options of each control and indicator are taken into consideration during web page generation process.

For example, appearance of Boolean control and indicators will be one-to-one correspondence in the LabVIEW front panel and on the web-page (Apendix A, figure A1). Positions, dimensions and sizes of each GUI element will be preserved. Moreover, changes made by a user will also be taken into consideration, for example colors of "on" and "off" states of the LED indicators. All LabVIEW user interface themes (or styles) will be demonstrated – "Modern", "Classic", "Silver", "System" and a few others. Other complex GUI objects are also supported by WebPager out-of-box. For example, sliders (Appendix A, figure A2) will keep their appearance and will correspond to options chosen for them by user in LabVIEW IDE. Plots are also presented featuring any number of them per application and user-defined options such as colors, line width, bar plots, fill base line and points' style (Appendix A, figure A3). Different fonts, colors, styles and sizes for labels, captions and free labels are also shown (Appendix A, figure A4).

WebPager tool from Labicom.net can be used for rapid deployment and sharing remote laboratories based on NI ELVIS boards. Virtual instrument controlling NI ELVIS can be put into the web-page in a few clicks.

WebPager is a certified by National Instruments software. It is used by industry but Labicom.net provides free licenses for remote labs hosted on the platform.

IV. CONCLUSIONS

The WepPager tool used by Labicom.net online laboratories platform is able to automatically generate pluginless web-pages rendered in HTML5 enabled browsers. The page is created on-the-fly without any modifications to original LabVIEW code. It also provides real-time user interaction through standard Internet protocols. WebPager can be used in remote laboratories for rapid prototyping and testing and in simple remote labs. Also it can be used for deploying NI ELVIS remote labs on the web.

REFERENCES

[1] Remote Panels in LabVIEW – Distributed Application Development [Online]. Available: http://www.ni.com/white-paper/4791/en (retrieved 19.11.2013)

[2] LabVIEW Web UI Builder Overview [Online]. Available: http://www.ni.com/white-paper/11602/en (retrieved 19.11.2013)

[3] H. Vargas, J. Sánchez Moreno, C. Salzmann, F. Esquembre and S. Dormido et al. "Web-Enabled Remote Scientific Environments", in Computing in Science and Engineering, vol. 11, num. 3, pp. 36-46, 2009. http://dx.doi.org/10.1109/MCSE.2009.61

[4] Doru Ursutiu, Cornel Samoila and John Bergmans, "LabSocket Technologies in Remote Engineering and Education" *in Proc. IEEE EDUCON 2013 conference*, March 13-15, 2013, Berlin, Germany, pp. 1272-1275.

[5] WebPager – Put LabVIEW front panels in the browser [Online]. Available: https://labicom.net/labdeveloper/WebPager/ (retrieved 18.11.2013)

[6] I. Titov, O. Smirnova, A. Glotov and A. Golovin "Remote Laser Laboratory: lifebuoy for laser engineering curriculum (a safe and cost-effective solution for hazardous lab environment)" // International Journal of On-line Engineering (IJOE), May 2012, vol. 8, issue 2, PP. 23-27 DOI: http://dx.doi.org/10.3991/ijoe.v8i2.1761.

[7] Igor Titov, Evgeny Titov "Labicom.net – Putting your laboratory online in less than five minutes with WebPager tool: Automatic generation and real-time control of a LabVIEW based laboratory server from pluginless HTML page.", Proceedings of the International Conference on Remote Experimentation (ExpAT'2013), Coimbra, Portugal, 18-20 September 2013, pp. 181-184

[8] Igor Titov "Labicom.net – The on-line laboratories platform: optimal solution for deploying cost-effective remote and virtual laboratories", IEEE Xplore, Proceedings of IEEE Global Engineering Education Conference (EDUCON), 2013 IEEE, Berlin, 13-15 March 2013, PP. 1137-1140, http://dx.doi.org/10.1109/EduCon.2013.6530251

[9] Labicom.net – Real labs at YOUR home [Online]. Available: https://labicom.net/ (retrieved 18.11.2013)

[10] The WebSocket Protocol – Internet Engineering Task Force (IETF) Request for Comments 6455 [Online]. Available: http://tools.ietf.org/html/rfc6455 (retrieved 18.11.2013)

AUTHORS

Igor Titov is with Labicom.net – the online laboratories platform and Bauman Moscow State Technical University (BMSTU), Moscow, Russia

Evgeny Titov is with Bauman Moscow State Technical University (BMSTU), Moscow, Russia

This article is an extended and modified version of a paper presented at the International Conference exp.at'13, held 18-20 September 2013, in Coimbra, Portugal. Submitted 18 November 2013. Published as resubmitted by the authors 04 December 2013.

APPENDIX A. COMPARISON OF LABVIEW FRONT PANELS AND WEBPAGER GENERATED BROWSER BASED PAGES.

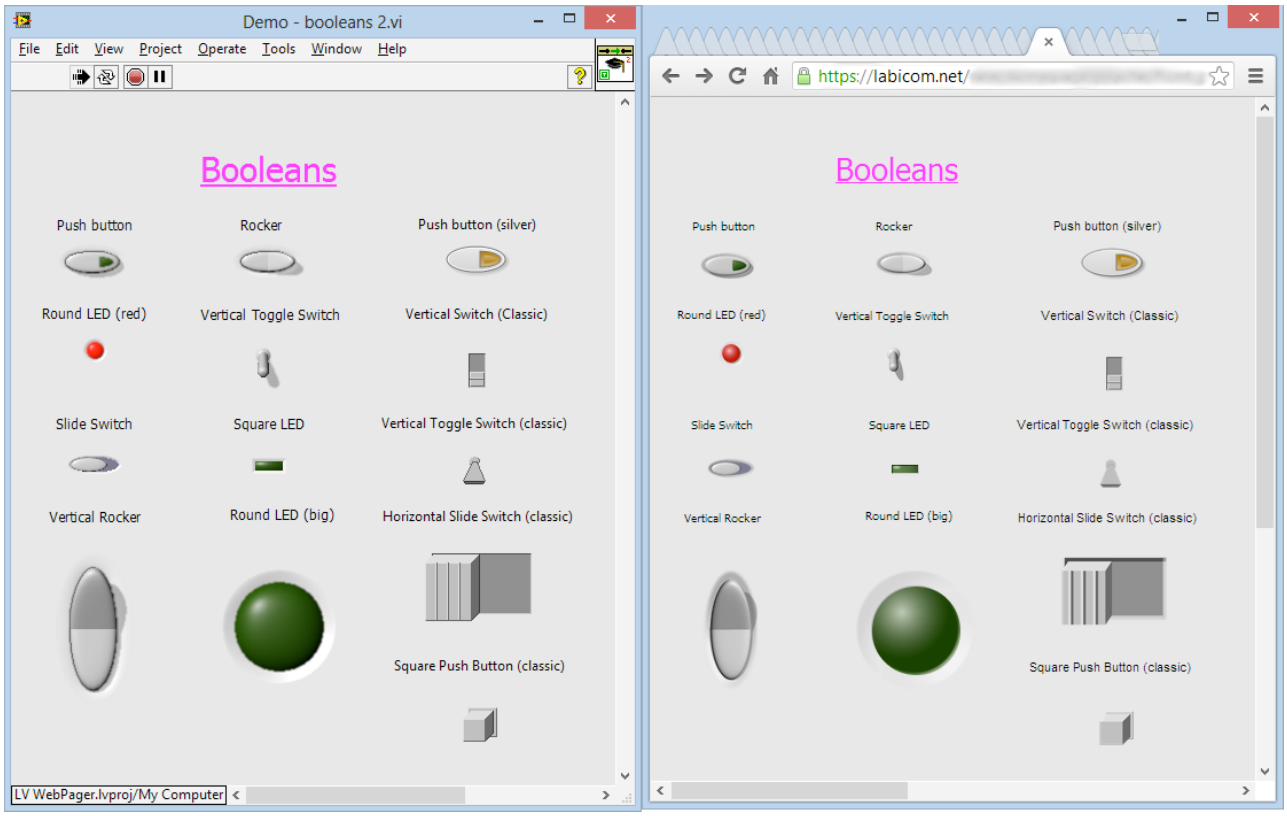

Fig. A1 – WebPager web-page generation with boolean controls and indicators.
On the left LabVIEW front panel, on the right – one-to-one web-page in the browser generated by WebPager.

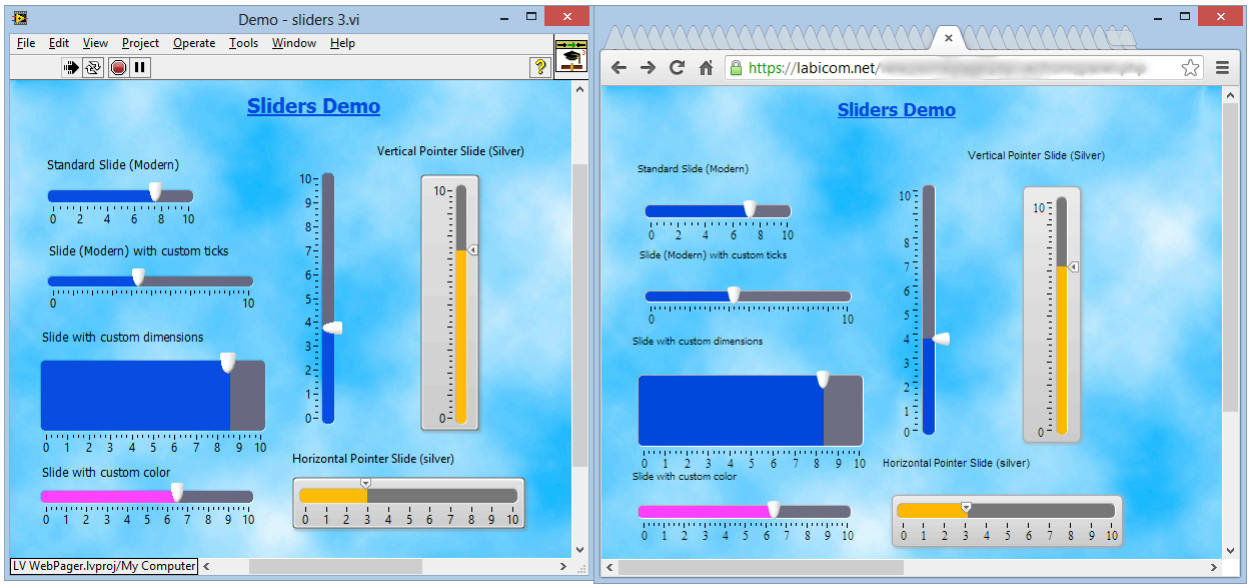

Fig. A2 – WebPager web-page generation with slider controls.
On the left LabVIEW front panel, on the right – web-page in the browser generated by WebPager.

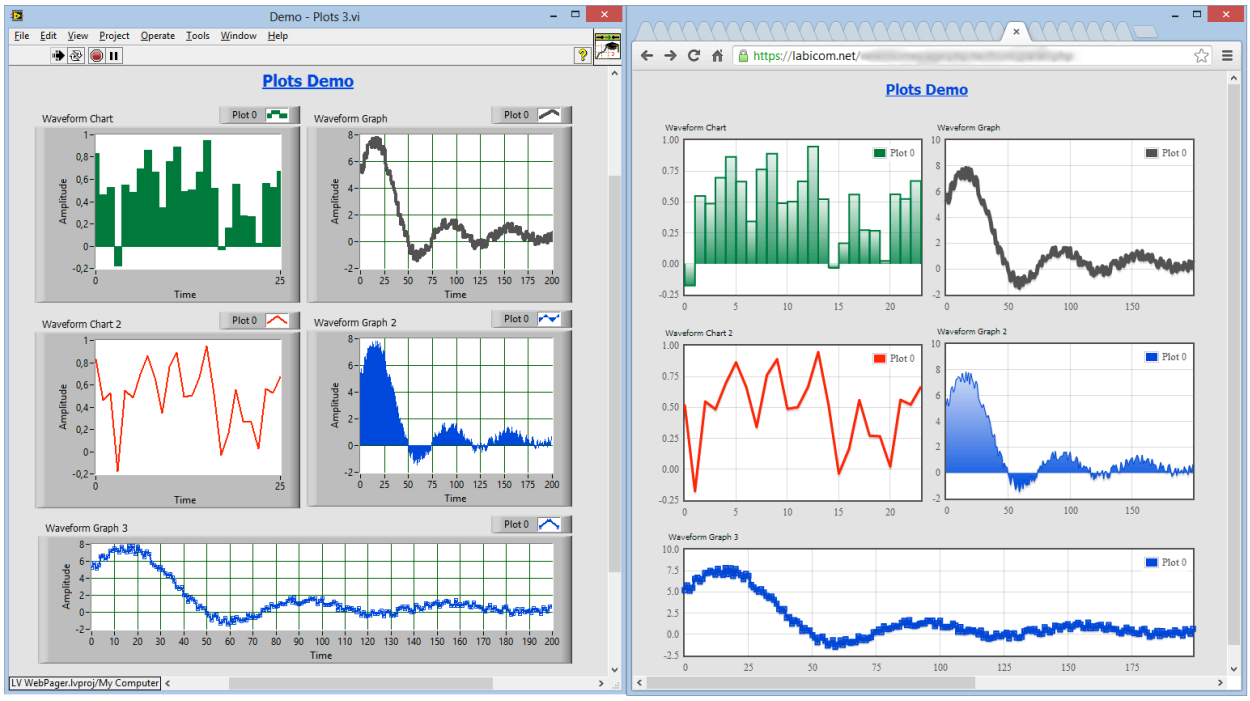

Fig. A3 – WebPager web-page generation with waveform graphs and waveform charts.
On the left LabVIEW front panel, on the right – web-page in the browser generated by WebPager.

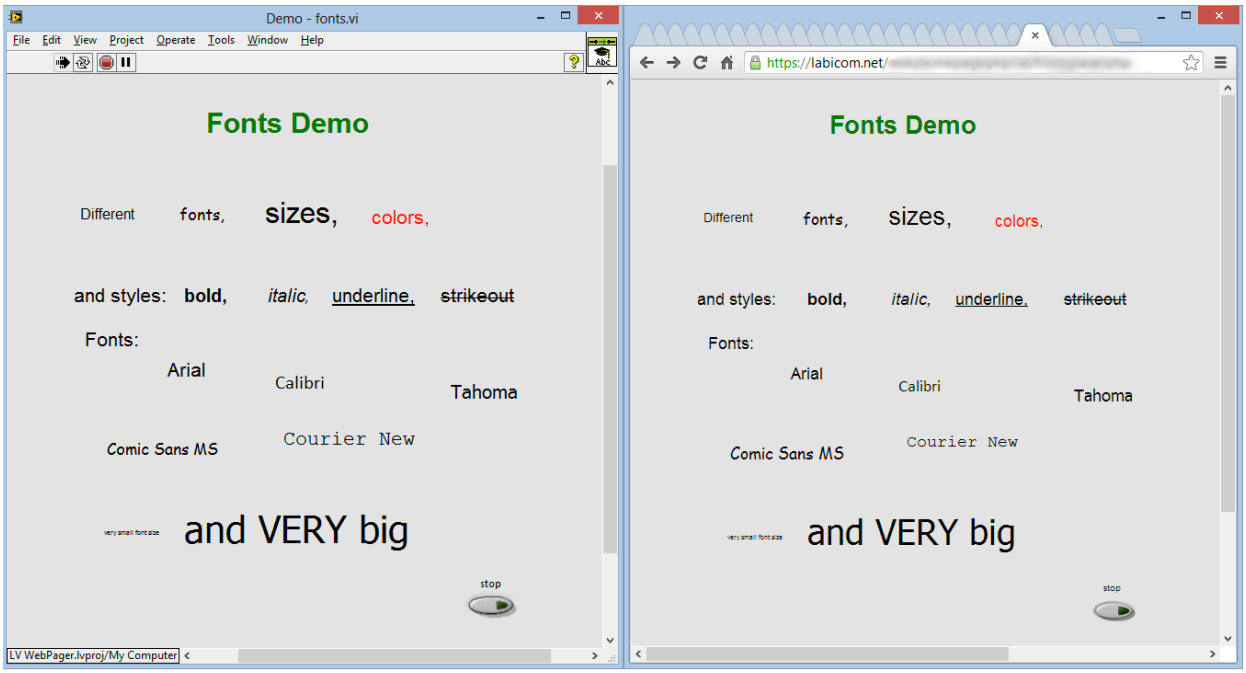

Fig. A4 – WebPager web-page generation with free labels of different fonts, colors, sizes and styles.
On the left LabVIEW front panel, on the right – web-page in the browser generated by WebPager.