

## **Survey of LabVIEW Technologies for Building Web/Internet-Enabled Experimental Setups**

Masoud Naghedolfeizi, Sanjeev Arora, and Singli Garcia

Fort Valley State University

### **Abstract**

The impact of World Wide Web on education has gone beyond the text and multimedia based instruction in course offering through the Web. Today, a number of universities, national laboratories and companies are using Web/Internet-enabled applications that can be fully controlled and monitored from remote locations. Continuous advances in computers and electronics coupled with falling prices in these industries have made Web/Internet-based technologies less costly than before, particularly for educational organizations. Thus, it is more affordable to invest in these technologies that are essential for both expanding education over Web and further improving and advancing such technologies.

The LabVIEW software from National Instruments company has significantly helped researchers and educators to integrate Internet/Web with experimental setups in various methods. Some of these methods have been also improved and further advanced by other companies to greatly facilitate the implementation of Web/Internet-enabled technologies. This paper presents a survey of Web/Internet-enabled technologies to build experimental setups that can fully be operated, controlled and monitored remotely. Both advantages and disadvantages of each of these technologies are discussed and evaluated.

### **Introduction**

In recent years, there has been an increasing interest in many institutions of higher learning to offer various online courses including those with laboratory components. These courses are mainly designed to compliment and enhance the traditional method of education in general and to provide students with a new and flexible method of learning in particular. The fact that the Internet/Web is currently being utilized extensively in distance-learning<sup>1</sup> shows that educational organizations have made a strong commitment to distance learning through Internet and World Wide Web. Today, many colleges and universities are offering on-line degree programs and/or courses in record numbers. However, a major concern regarding on-line courses with laboratory components has been and still is the practicality of bringing laboratory experiments on-line. In this regard, the complexity associated with interfacing technologies between a remote site and a local site running an experimental set-up poses a major technological challenge in this area. Perhaps, much of this complexity stems from non-

availability of suitable software systems that could harness the power of Internet/Web to smoothly run and control experimental applications. LabVIEW, a leading computer-based measurement software system, from National Instruments company<sup>2</sup> has been used by many researchers and educators around the world for full automation and data acquisition of complex engineering systems. In fact, a few colleges, universities and private educational organizations have already used the software from remote sites to interact with applications running on local sites through Internet/Web<sup>3</sup>. Since LabVIEW offers a number of different methods to operate, monitor, and/or control an application (such as an experimental setup) via Internet/World Wide Web, it is important to understand the range of applicability of each method and its implementation challenges in practice. It is also important to know that a certain level of knowledge in both LabVIEW programming and principles of Internet technologies is required to utilize these methods.

This paper discusses the major methods of interfacing LabVIEW applications with the Internet/WWW that could be employed for remote operation of experimental setups. The paper primarily focuses on advantages and disadvantages of each method solely from educational point of view. Typical experiments that could be supported by each method are also examined.

### **On-line Technologies in LabVIEW**

The two categories of LabVIEW applications with online capabilities are WWW and Internet. The WWW-enabled LabVIEW technologies could be very simple or relatively complex depending on the type of application or the use of adds-on software systems. The major Web methods supported by LabVIEW include:

- Remote Viewing using LabVIEW Built-in Web Server
- CGI technology
- DataSocket and ActiveX
- DataSocket and Java applet
- AppletVIEW

The Internet methods apply TCP/IP protocol to build Internet-enabled applications. Similar to the Web methods the complexity of Internet methods depends on the application type and the use of adds-on software systems. It should be noted that applications using the Internet methods are based on Client/Server architecture and are not generally controlled through the Web. These applications normally require the remote site (client) to have a LabVIEW software (client program) to access the server site that contains the main application running for example an online experiment. The major Internet methods include:

- Client-Server using LabVIEW built-in TCP/IP functions
- VI Server
- DataSocket

### **Remote Viewing using LabVIEW Built-in Web Server**

This is the simplest method of interfacing a LabVIEW application with the World Wide Web. In this method, panel images of the application can be constantly broadcast to users through the Web at relatively short time intervals. An advantage of this method is its simplicity and the fact

that it does not need any coding or adds-on software systems. However, a main disadvantage is that the user cannot interact with the (Virtual Instrument) VI. Also, a relatively notable slow-down in the execution of application may occur if a large number of users co-currently access the application. The configuration of the LabVIEW built-in Web Server is very simple and straightforward as thoroughly described in reference<sup>4</sup>.

This technology may be used for experiments that only need fixed input values and can be easily repeated at certain time intervals in automatic or manual manner. It should be pointed out that this method does not display data in real time and the remote user does not have a direct access to the data generated during the application run. To apply this method for an online experiment, the LabVIEW program of the experiment must contain routines to post the experimental data on the Web site of the experiment (or the course related to the experiment) and inform the user when the experimental run ends. This method allows students to observe the experiment and later use the experimental data accessible through the Web for analysis. At Fort Valley State University, this technology has been used to offer RC experiments on-line.

It is also important to point out that since the built-in Web server was added to the LabVIEW version 5.1 and above, students and educators using earlier versions need to upgrade the software.

### **CGI Technology**

CGI (Common Gateway Interface) essentially is a standard for the construction of dynamic Web pages by external programs installed on a Web server<sup>5</sup>. In other words, CGI simply defines an interface protocol by which a Web server communicates with other applications. CGI scripts could be written in nearly any programming language including LabVIEW<sup>4</sup>. The Internet Toolkit for LabVIEW<sup>6</sup> which is an adds-on software package from National Instruments, supports CGI applications in form of Virtual Instruments (VIs) that could be used to develop interactive applications for Web-enabled experimental set-ups.

CGI technology has proven to be relatively powerful tools for deploying Web-enabled applications<sup>7</sup>. In practice, the CGI technology broadcasts the imagemap of a LabVIEW application over the Web at very short time intervals. The image map normally contains areas (hotspots) in form of buttons and/or switches for the Web user to interact. However, this technology cannot support extensive user interaction since it requires a more detailed imagemap that corresponds to an increased reloading time of the Web page. Similar to the built-in Web server, the data cannot be broadcast in real time and the Web user does not have a direct access to the data during the execution of the application. The server that hosts the application may publish the data on the Web after the application execution ends.

Employing CGI technology is generally complicated since the developer needs to have a good understanding of CGI technologies as applied in LabVIEW as well as HTML coding that utilizes CGI tags to build dynamic Web pages. The other disadvantage of this technology is the fact that the developer needs to purchase additional software system to design a Web-enabled application.

This technology is basically suitable for designing lab experiments that need moderate user interaction. This technology was employed in experiments such as RLC and unsteady heat conduction measurements at Fort Valley State University. Through the Web, a student is able to

turn on the physical experiment and change values of parameters such as frequency and heat flux. A potential problem with this technology could be server crashes due to the execution of CGI programs that are not well designed.

References 4 and 7 provide a good description of this technology with ample examples for a beginner who has background in LabVIEW, CGI programming, and HTML coding.

### **DataSocket and Active X**

DataSocket is a new technology developed by National Instruments to greatly facilitate live data exchange between LabVIEW based applications over the Internet/Web without any TCP/IP programming<sup>8</sup>. The combination of data socket and ActiveX technologies could be used to build interactive LabVIEW Web-based applications. Such applications may be used to broadcast and receive live data and to support a greater user interaction. An attractive feature of this technology is in-place data manipulation that can be performed in the ActiveX controls and thereby improving the raw data transmission rate by reducing the data processing load in the LabVIEW application. However, the development and implementation of such applications is rather involved. The developer needs to build ActiveX controls using Visual Basic and ComponentWorks from National Instruments and then embed them in an HTML file. Thus, the developer should be familiar with Visual Basic programming, HTML coding and LabVIEW programming. Some other inconveniences associated with this technology are as follows:

- Only works with Microsoft Internet Explorer browser.
- The user needs to change security settings of Internet Explorer in order to enable ActiveX controls and then reboot his/her computer in order the changes to take effect.
- The user may not see future upgrades to the application on his/her computer that has been used to see the older version of the application.
- The developer must purchase the ComponentWorks software system from National Instruments.
- ActiveX controls may cause security threats for client computers.

We have utilized this technology to monitor and interact with a LabVIEW application employed to gather meteorological information in our lab. Though there was a major improvement in observing data on a Web browser as compared to the previous methods, we did not find this technology particularly user friendly in instances when the application needed frequent upgrades.

### **DataSocket and Java Applet**

DataSocket and Java Applet technology is very similar to ActiveX technology in many respects. This method uses Java Applets to interface a LabVIEW application with the Web. To apply this technology, the application developer needs to have some background in Java programming and building Java Applets that can be embedded in an HTML file for the Web viewing. The use of visual development environment (such as Visual Age<sup>9</sup>) can greatly facilitate building Java Applets that need minimum or no Java programming. The advantages of this technology over ActiveX are as follows:

- The Java Applet development software system such as IBM Visual Age is often inexpensive.
- The Applets are viewable on both the Netscape and Internet Explorer browsers.

- The client computers do not need to change security settings on their browsers or to reboot their computers to initialize the Applets.

However, as was the case with ActiveX technology, the developer must purchase a graphical instrumentation controls (known as JavaBean Controls) that can be used to build a more complex Web interface containing items such as knobs, charts, switches, and gauges. The ComponentWorks from National Instrument currently does not have any JavaBean controls. These controls could be purchased from other vendors such as Ergo Tech<sup>10</sup>.

This technology has not been used as yet at Fort Valley in developing an online experiment; however, our experience with this technology for a simple signal generating simulation application was very encouraging due to ease of application development and convenient upgrades.

### **AppletView**

AppletView from Nacimiento Company<sup>11</sup> is a new technology developed to design Java Applets that interface with LabVIEW. Using this technology allows the application developer to design Applets without any Java programming and other requirements mentioned in the previous section. Furthermore, AppletView uses its own TCP/IP protocol (not DataSocket) for data communications over the Internet or Web. AppletView is also viewable on both the Explorer and Netscape browsers and is considered to be a multi-platform application. The other advantage of the AppletView is that it operates based on LabVIEW programming structure and therefore, the developer does not need to know any additional programming language.

Using AppletView in practice is very simple and its User Guide<sup>11</sup> provides adequate information with ample examples that show how to work with the software. At Fort Valley, AppletView has been employed to build a simulation experiment to display three waveform signals namely; Sine, Square, and Toothsaw with variable frequency and a fixed noise. Also, the online experiments developed in previous technologies are currently being converted to AppletView technology. It should be noted that AppletView is a very new technology and it is being improved and upgraded rather frequently. The graphical appearance of instrumentation controls on the Web may not be as impressive as those in LabVIEW.

### **Client/Server Model using LabVIEW TCP/IP Functions**

As mentioned earlier, this technology is used to develop Internet-enabled applications. In this model, both the server and client applications use LabVIEW software and thus there is no need for any additional programming or adds-on software systems. Users of this model should possess a relatively good knowledge of Internet programming in general and TCP/IP protocol in particular. The two general categories of this model are single-client and multiple clients. In practice, the applications designed under either of these categories are rather involved, but they could be even more complicated in multiple-clients class where the server application needs to handle and manage multiple clients sessions at the same time. Fortunately, in performing an online experiment, it is generally desirable to have a single client session due to the nature of

experiment that needs only interactions from a single user. A requirement in implementing this technology is that the server application must be running at all times so that it can take client calls.

Since this method supports a low level TCP/IP programming, the developer has more flexibility in designing the application. For example, the developer may include a routine in the server application that notifies other client sessions that the server is already in use by another client. It may also include a routine that places incoming client sessions in a queue to interact with the server application. The inclusion of these routines in the design of an online experiment is particularly important to resolve the problem of co-current users trying to run the same experiment.

The client-server sessions are generally secure since only authorized users have the client applications and the host name or IP address of the server machine. The client applications should be distributed in form of executable files if installing LabVIEW software on the client machines becomes expensive. However, these executable files tend to be very large. It should be noted that the maintenance of client applications could be expensive and time consuming since upgrades to the server applications may require changes in the client applications and this in turn results in the redistribution of new client applications.

This technology was used at Fort Valley to build two sophisticated on-line experiments; namely, a motor-generator experimental setup and a flow-level station<sup>12</sup>. The server application on the computer interfaced with the each experiment runs at all times and the client application interacts with the server application with control instruments such as knobs, switches and sliders. The generated experimental data is sent to the client when the experimental run ends. The client is also able to monitor all the measured experimental parameters such as motor and generator voltages and currents, applied torque on the generator, various temperatures, flow rate, pressure, and fluid level. A detailed description of these experiments as well as feedback received from students who used the experiments is provided in an earlier paper<sup>12</sup>.

## **VI Server**

VI server is a feature added to LabVIEW version 5.1 and above to easily build client server applications over TCP/IP for networked computers. In this method, the developer does not need to know Internet programming or how to use TCP/IP functions in LabVIEW. However, the developer needs to know the principles of object oriented programming in LabVIEW. In the VI server technology, the developer requires to define Application and/or Virtual Instrument (VI) classes<sup>4, 13</sup> to design client and server applications. For example, Application Class is used when there is a need to obtain information about the graphical programming environment such as the type of operating system and application directory path. The VI class is employed to set or get properties of a VI, print the VI, and/or run the VI. The VI class method is often used for designing online experiments.

The client server applications on networked computers are functional when the VI server is enabled. The VI server configuration dialog box allows the developer to enable the server and set other properties of the server such as port number and TCP/IP access privileges needed for

security. The VI server configuration is straightforward and described in detailed in references 4 and 13.

This technology requires LabVIEW running on both client and server machines. It is possible to make executable files of client and server applications but TCP/IP port setting should be changed to some non-default values before building the executable files.

Similar to the technology mentioned in the previous section, the maintenance of applications in VI server method could be both expensive and inconvenience. In addition, both the client and server machines should run the same version of LabVIEW to avoid problems arising from version incompatibility.

This method was utilized to control and monitor a AA-battery performance testing experiment at Fort Valley. Students through the Internet and VI server technology are able to turn the experiment on or off, monitor the trends of battery voltage and current with time, and record measurements of the experimental parameters.

### **DataSocket**

As described earlier DataSocket technology is mainly used to exchange live data between local and remote sites. This technology includes two components: the DataSocket server application and DataSocket client application. The DataSocket server publishes data and manages client connections. The DataSocket client application is used to read and manipulate the published data. In networked computers or Internet connectivity both the server and client applications are written in LabVIEW.

The DataSocket server is very easy to configure and run as described in reference<sup>8</sup>. The design of a DataSocket server and client applications is straightforward and described in references 4 and 8 through a number of easy to follow examples. The main advantage of this method lies in its simplicity. Unlike the two previous methods, the developer does not need to have a background in Internet programming, TCP/IP or object oriented programming concepts. Since this method is mainly used to receive live data from a server machine, co-current users can be easily managed without any extra programming or invoking complicated routines in the server application.

The DataSocket method can also handle bi-directional communications between client and server applications. However, extensive data exchange between client and server may result in notable application slowdowns. In addition, managing of co-current users in bi-directional communications becomes more challenging since when a client is using the server the other clients should not be able to interact with the server.

We have used this method to generate simulated Sine, Square, and Toothsaw signals in the server application that broadcasts these signals to a client application. The client application includes control instruments (an on/off button and a drop menu) and charts to operate the experiment as well as select and monitor a desired signal. Our applications were not designed to handle co-current users.

Table 1 lists some of the salient features of all Web/Internet enabled technologies using LabVIEW software system.

Table 1: Advantages and disadvantages of LabVIEW technologies for Web/Internet enabled applications.

	<b>Application Development</b>	<b>Range of Applications</b>	<b>User Interaction</b>	<b>Programming Knowledge</b>	<b>Security</b>	<b>Cost (additional software)</b>
<b>Remote Viewing*</b>	Very simple	Limited (no live data)	None	Only LabVIEW	Good	None
<b>CGI Technology*</b>	Relatively difficult	Moderate (no live data)	Moderate	LabVIEW +CGI +HTML	Good	Adds-on Internet toolkit
<b>Data Socket and ActiveX*</b>	Moderately difficult	Good; (supports live data)	Good (only on Explorer)	LabVIEW + Visual Basic + HTML	Could be problematic	Component Works +VB
<b>Data Socket and Java Applet*</b>	Minimally difficult	Good (supports live data)	Very good	LabVIEW+JAVA+HTML	Good	Visual Java + VIB
<b>Applet View*</b>	Relatively simple	Good (supports live data)	Very good	LabVIEW +AppletView	Good	AppletView
<b>Client-Server (TCP/IP)**</b>	Difficult	Very good (supports live data)	Excellent	LabVIEW	Very Good	None
<b>VI Server**</b>	Relatively difficult	Good (supports live data)	Good	LabVIEW	Very Good	None
<b>Data Socket **</b>	Minimally difficult	Good (supports live data)	Good	LabVIEW	Good	None

\* Used in Web Enabled Applications

\*\* Used in Internet Enabled Applications

VB Visual Basic

VIB Virtual Instrumentation Beans

### Typical Student Comments

Students using some of the above technologies were surveyed to obtain some feedback regarding the effectiveness of on-line laboratory experiments. Some of the comments resulting from the survey are provided below:

- As lab group members, we can meet at times suitable to everyone to run the labs instead of being required to be here for a set class time
- The ability to perform experiments from remote locations was interesting and fun.
- You can run experiments during class time or at 3AM as well as turn in your reports at any time over the web
- I liked the freedom of the lab



- It worked out well being able to work at different times
- This was a new experience. Especially running the experiments on the Web, and gathering the experimental data through the Web.
- ActiveX Web-based experiments were not very convenient to use since the computer had to be rebooted.
- Web access learning is very important these days, when a lot of information is available and e-mail is taking the place of traditional phone calling and letter writing.

Some major concerns were network congestion, non-availability of the remote site, and long queues due to too many co-current users.

## Conclusions

This paper briefly discussed the available technologies in LabVIEW to build an experiment that could be controlled, monitored, and/or operated through the World Wide Web or Internet. The discussion mainly focused on the advantages and disadvantages of each technology. It is obvious that the type of an experiment, the knowledge level of the developer in LabVIEW and the Internet/Web, and cost are major deciding factors in selecting one method over another. Factors such as operational security and safety in an online experiment could also play an important role in choosing one of these technologies.

The development of new technologies such as VI server, DataSocket, and AppletView indicate that that National Instruments and other companies are constantly improving upon Internet technologies in LabVIEW or adds-on software systems to make the integration of the Internet/Web in LabVIEW applications as easy and smooth as possible. Therefore, it is evident that as these technologies evolve and become more affordable, educational organizations will be able to offer more online courses that include a lab component.

## Acknowledgements

This work was made possible through a grant entitled “Developing a Minor Program in Computer Based Measurement and Instrumentation at Fort Valley State University” awarded under the NASA Curriculum Improvement Partnership Award Program (CIPA).

## Bibliographic Information

1. Sandra Kerka, “Distance Learning the Internet and the World Wide Web,” Eric Digests, ED395214, 1996.
2. National Instruments Web Page: [www.ni.com](http://www.ni.com)
3. Chemical Engineering Department, University of Tennessee at Chattanooga: <http://chem.engr.utc.edu>
4. Jeffrey Travis, "Internet Applications in LabVIEW," Prentice Hall, 2000.
5. Mark Felton, “CGI: Internet Programming in C++ and C,” Prentice Hall, 1997
6. National Instruments, "The Measurement and Automation Catalog 2001," [www.ni.com](http://www.ni.com)
7. National Instruments, “ Internet Developers Toolkit for G: Reference Manual,” [www.ni.com/support/labview/toolkits/internet](http://www.ni.com/support/labview/toolkits/internet)

8. National Instruments, "Integrating the Internet into Your Measurement System: DataSocket Technical Overview," [www.ni.com/Internet](http://www.ni.com/Internet)
9. [www.software.ibm.com/ad/vajava](http://www.software.ibm.com/ad/vajava)
10. [www.ergotech.com](http://www.ergotech.com)
11. [www.nacimiento.com](http://www.nacimiento.com)
12. M. Naghedolfeizi, S. Arora, and J. Henry, "Remote Laboratory Operation: Web Technology Successes," American Society for Engineering Education (ASEE) Annual Conference & Exposition National Meeting, Albuquerque, NM, June 24-27, 2001.
13. National Instruments, "G programming Reference Manual," Austin, TX, 1999.

## **Biographical Information**

- ***Masoud Naghedolfeizi***

Dr. Naghedolfeizi is an associate professor in the Department of Mathematics and Computer Science at Fort Valley State University. He completed his Ph.D. in engineering from the University of Tennessee at Knoxville. He is actively involved in developing modern computer based laboratories at Fort Valley State University. His research interest includes computer-based measurement and instrumentation, applied artificial intelligence, and computerized tomography.

- ***Sanjeev Arora***

Dr. Arora is an associate professor in the Department of Mathematics and Computer Science at Fort Valley State University. He completed his Ph.D. in Physics from the University of Delaware. He is interested in using computer based instruction techniques in the physics courses offered at Fort Valley State University

- ***Singli Garcia-Otero***

Dr. Garcia-Otero is an associate professor in the Department of Mathematics and Computer Science at Fort Valley State University. She completed her Ph.D. in Engineering from the University of Missouri. She is interested in digital signal processing, biosensors, computer-based measurement and instrumentation and parallel processing.