Design of a Web-Based Virtual Laboratory Instrument Measurement Interface

Hamadou Saliah-Hassane[1], Patrick Dumont-Burnett[2], Christian Loizeau[3]

Abstract *– This paper presents a Web-based measurement instrument interface model and demonstrates its usefulness for educational purposes. We will present in detail how we built an interface designer, a software tool that presents a canvas on which controls are dropped from a list. These controls can then be moved and resized and their properties set through a property window. The binding of these controls with their data source, in this case, from programs running on remote computers, is also made through this property window. The interface designer can switch at any time from the design mode to a test mode to verify the behavior of the interface in real-time. A user interface can then be designed by dragging and dropping the required graphical components, then, at the end of the whole process, by clicking on a button that generates a corresponding XML file. This file is then sent to a Web server to be read by an ActiveX control displayed on a Web page we have called PolymorphiX. A live demo of the whole virtual laboratory environment will be presented.*

Index Terms *– distributed measurement system, ergonomic user interface, laboratory innovation, virtual laboratory.*

INTRODUCTION

When geographically dispersed users are collaborating or cooperating on performing a task over a network via a computer system and are linked to measurement instruments integrated within a e-learning environment [1]-[3], it is important that the manager of the over-all environment have efficient and quick-performance tools to generate customized user interfaces. Thus, each participant involved in a synchronous task, depending on the role s/he has been assigned, can interact with the other actors and with measurement instruments in a safe and secure way. The Web interface generator we are presenting is an essential and very useful tool for managing the activities taking place in virtual and remote laboratories and for automated systems utilized in testing equipment via Web interfaces.

The interface we are proposing is made up of three components that we shall describe in the sections that follow. To design this interface as well as the over-all system environment, we used commercial products from National Instruments such as LabVIEW 6.0, ComponentWorks 3.0 and the following Microsoft products: Visual Basic, Access 2000, Active Server Pages 3.0, IIS5 on Windows 2000 Server and Internet Explorer 5 [5]-[6].

THE INTERFACE DESIGNER COMPONENT

The first component is an application called Interface Designer that makes it possible to design and test customized interfaces that allow users to interact with remote measurement instruments and with other users via the Web. In fact, at this stage of our project, the graphical interfaces thus created serve as an interface with measurement instrument interfaces developed with LabVIEW; applications developed with LabVIEW are called Virtual Instruments (VI). Controls for both these interfaces are linked via a DataSocket server and communicate via the DSTP protocol (Figure 1); DataSocket Transfer Protocol is a very useful protocol developed by National Instruments that is becoming standard.

[1] Hamadou Saliah-Hassane, Télé-université, LICEF/CIRTA Research Center, 4750 Henri-Julien, Montréal, H2T 3E4 saliah@licef.teluq.uquebec.ca
[2] Patrick Dumont-Burnett, LICEF/CERTA Research Center, 4750 Henri-Julien, Montréal H2T 3E4 pdb1@hotmail.com
[3] Christian Loizeau, Collège de Maisonneuve 3800 Sherbrooke St. East, Suite B-2250, Montréal H1X 2A2 cloizeau@videotron.ca

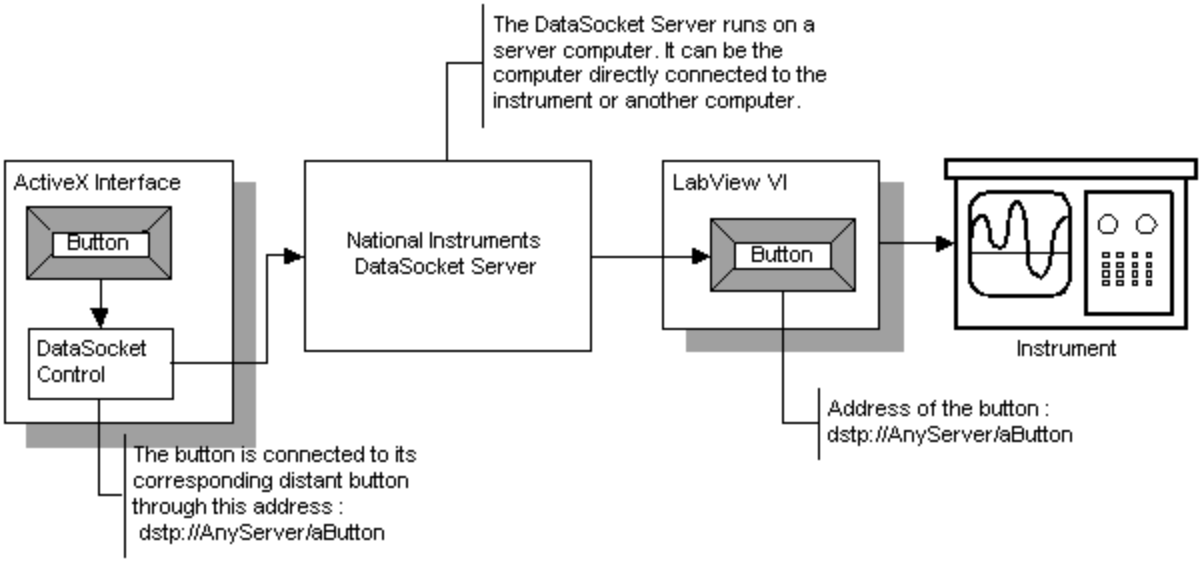

FIGURE 1 INTERFACE LINK WITH THE REMOTE MEASUREMENT INSTRUMENT.

The Interface Designer displays a canvas on which controls are dropped from a list and can be moved and resized. These controls are set through a property window. The interface designer can switch at any time from the design mode to a test mode to verify the behavior of the interface (Figure 2).

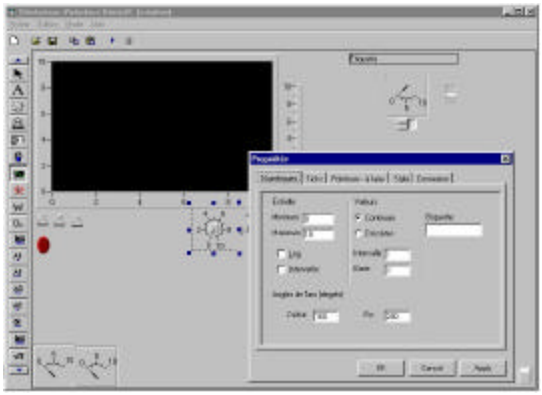

FIGURE 2 INTERFACE DESIGNER.

Once the user interface graphical design process is completed, an XML (eXtended Markup Language) file is automatically generated by clicking on the mouse. This file is then sent to a Web server to be read by the PolymorphiX, a Web page ActiveX control. While the XML file is being generated, temporary files containing computer code control styles for reusable graphical components called ComponentWorks, are created on the designer's work station. These ComponentWorks controls have an export command to export their style to a file. File contents are immediately reread by the application that incorporates this data into the XML file. The style files are then destroyed. Another more elegant, but more laborious method is to read each property for each of the controls and to assign the result in the XML file, without going through the intermediate style files. This is the procedure that will be implemented in the end.

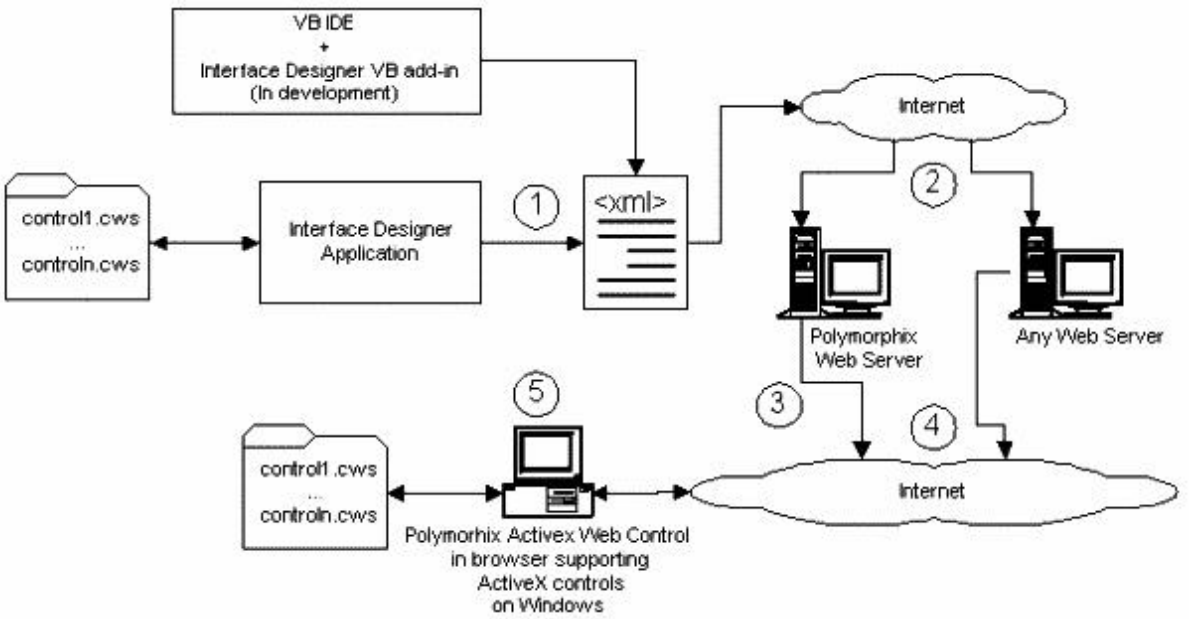

FIGURE 3 INTERFACE CIRCULATION WITHIN THE SYSTEM.

The interface circulation process (Figure 3) is described as follow:

1. An XML file containing inteface parameters is generated by the Interface Designer application. Styles associated with ComponentWorks controls used on the interface are exported as "cws" files to a temporary folder. Contents of these files are processed and incorporated with the XML file. The temporay folder is then deleted.
2. The XML file is transferred to a web server. The Polymorphix web server includes a special form for such transfer.
3. The end user downloads the Polymorphix ActiveX control from the Polymorphix web server which also verifies the user's identity.
4. The end user then requests for an interface (XML file) through the Polymorphix ActiveX control.
5. The interface is validated by the Polymorphix ActiveX control. Styles associated with ComponentWorks controls are extracted from the XML file and exported to files having "cws" extension into a temporary folder. These files are read by the corresponding controls in the user interface. The folder is the deleted.

The Interface Designer does not allow for adding a code. Programming codes only includes assigning values to properties via property windows. An additional component of Visual Basic (VB add-in) is currently being developed and will make it possible to generate the same type of property files as with the Interface Designer, but within a Visual Basic design development environment.

THE POLYMORPHIX COMPONENT

The PolymorphiX component is a Web-based ActiveX control. The user wishing to use the PolymorphiX component has to first go to the corresponding Web site. Once the ID validation process against a database through ASP scripts (Active Server Pages) is successfully completed, the user is directed to the Web page containing the component. Data about the user is transmitted to the component, again via ASP scripts, so that the laboratory controller and other users can identify the user. The user has to download the component on his or her work station during the first visit. The component is then installed on the user's work station. Thus, s/he need not download the component during future visits.

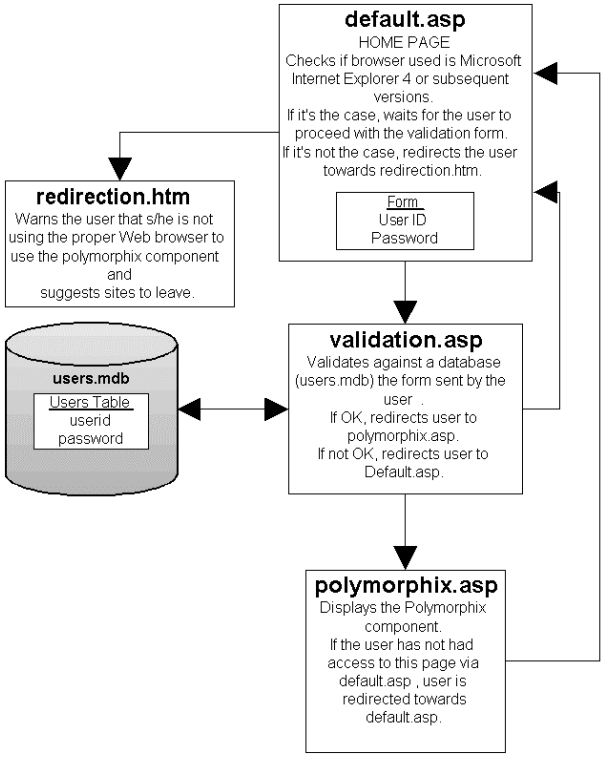

FIGURE 4 DATA FLUX ON THE POLYMORPHIX WEB SITE.

The control is initially presented in a graphical interface format with the upper half, which displays measurement instrument interfaces, empty. The lower half displays communication tools for the user controlling the laboratory task and for the other users involved in the experiment. Once the URL of the desired interface among the list of interfaces is entered or selected, the user executes the appropriate download. The PolymorphiX component recovers and displays the XML file corresponding with the desired interface.

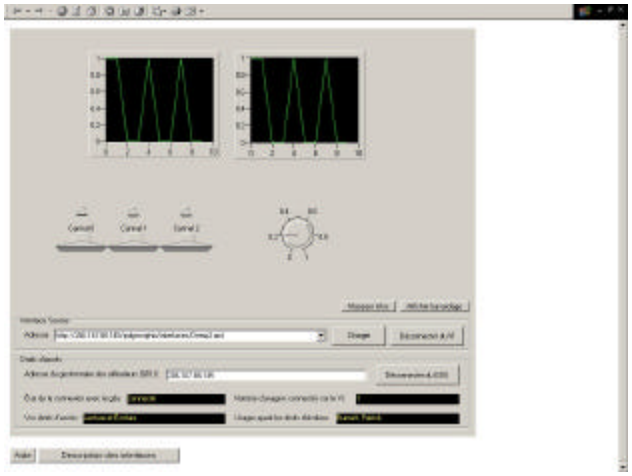

FIGURE 5 POLYMORPHIX COMPONENT DISPLAYING A MEASUREMENT INSTRUMENT INTERFACE.

THE USER ADMINISTRATION UTILITY COMPONENT

The User Administration Utility component is used to manage access to the remote measurement instrument. When the user downloads an interface within PolymorphiX, s/he can only receive data from the remote VI by default. The user can only send instructions to the VI once s/he has received authorization from the controller of the laboratory task via this component. This component receives the inquiries of various clients wishing to interact with the remote measurement instrument. The controller of the experiment thus has a list of users linked to the remote instrument and wishing to interact with the instrument and can thus manage user access. The component transmits data to users on the network regarding the user's access rights to the remote instrument (read-only access or access with property modifications), the number of users connected to the measurement instrument and the name of the user who has the right to modify the measurement instrument properties. This component also acts as a communications server, transmitting messages between participants involved in an experiment.

SYSTEM DEPLOYMENT

The user wishing to have access to the PolymorphiX component has to do so by using a Web browser that can support ActiveX control display. The PolymorphiX Web site was developed with Active Server Pages (ASP 3.0) technology and it was implemented on an IIS5 Web server running on Windows 2000 Server, all of which are Microsoft technologies or products. ASP technology can sometimes be run on operating systems other than Windows, but this has not been tested. The remote measurement instrument and its VI, as well as the DataSocket server, can be run on any platform supporting these technologies, but the User Administration Utility component, developed with Visual Basic, can only be run on a Windows OS.

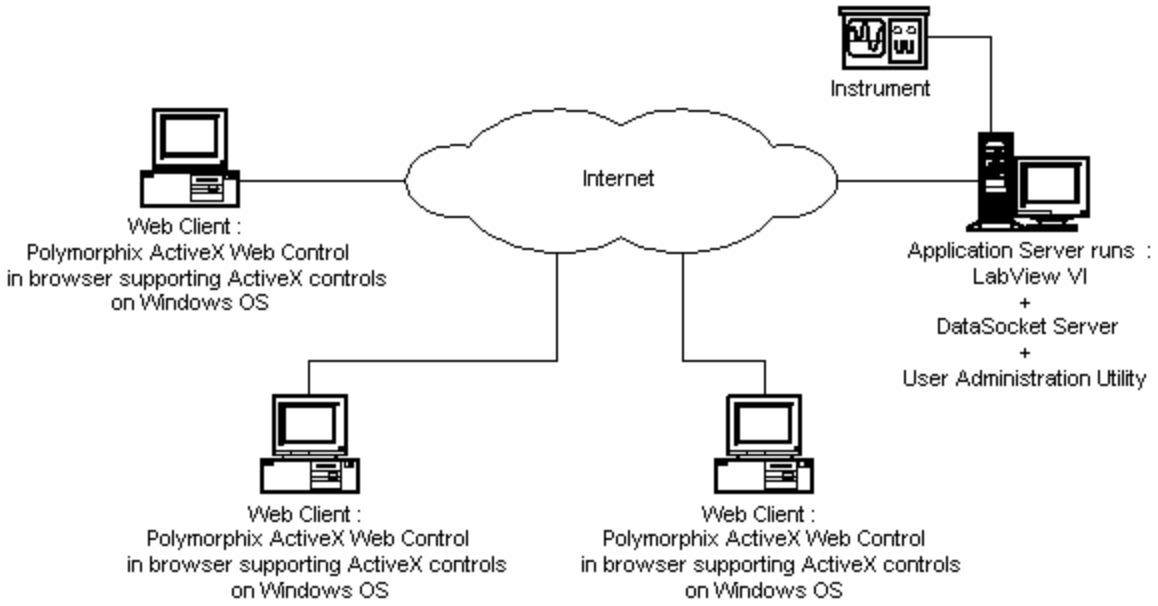

FIGURE 6 SYSTEM DEPLOYMENT.

CONCLUSION

In this paper, we have presented an overview of how we have designed an adaptable Web-based Virtual Laboratory Instrument Measurement Interface that allows geographically dispersed team members to collaborate and cooperate in carrying out laboratory tasks. The objective is to provide virtual or remote laboratory managers, teachers and learners with an easy-access work environment, depending on each participant's role in a synchronous activity, and depending on the task each has been assigned. In this paper, we have utilized mainly products from National Instruments. In the future, we shall incorporate other products such as Matlab [7], currently used in engineering training. We are also focussing our efforts on making these interfaces more intelligent, ergonomic and user-friendly.

AKNOWLEGMENT

The authors would like to thank the Canadian Innovation Fund and the Quebec University Network fund for their assistance in developping innovative e-learning environments and tools for teachnig in engineering.

REFERENCES

[1] G. Paquette, C. Ricciardi-Rigault, C. Paquin, S. Liégeois, E. Bleicher, "Developing the Virtual Campus Environment", *Proceedings of Ed-Media 96 World Conference on Educational Telecommunication*, pp. 244-249, Boston, USA, June 17-22, 1996.

[2] M. Bertocco, F. Ferraris, C. Offelli, M. Parvis, "A Client-Server Architecture for Distributed Measurement Systems", IEEE Transactions. on Instrumentation and Measurement, Vol. 47, No. 5, pp. 1143-1148, 1998.

[3] H. H. Saliah, A. Abecassis, E. Nurse, Design of a Generic, Interactive, Virtual and Remote Electrical Engineering Laboratory, *Frontier in Education Conference 99: Designing the Future of Science and Engineering Education*, San Juan Puerto Rico, November 10 – 13, 1999.

[4] A. Ferrero, V. Puiri, "A simulation tool fir virtual laboratory experiments in a WWW environnement", IEEE Transactions. on Instrumentation and Measurement, Vol. 47, pp. 741-746, 1998.

[5] LabVIEW User Manual, National Instruments, Austin, 2000.

[6] D. S. Platt, Introducing Microsoft® .NET, Microsoft Press, 2001

[7] The Mathworks Inc. "Learning Matlab", 2000