A Framework for a Distributed-Measurement User Interface

Hamadou Saliah-Hassane[1], Patrick Dumont-Burnett[2], Colombiano Kedowide[3]

Abstract *¾ In an earlier publication, we presented RVLabX, a remote virtual laboratory environment that lets many users interact while running lab sessions via Internet. This system is made up of virtual interfaces, measuring instruments and communication and control components. At the application program end, users and learners can interact by accessing these interfaces via a Web browser. At the server site, the teacher or his/her assistant can implement a software application that manages tutorial sessions within a virtual laboratory. The system's virtual instrument interfaces can be created either with the application we have developed to generate interfaces whose parameters are contained in the Extensible Markup Language (XML) files, read by a visualization control, or with a language compatible with ActiveX technology that implements a particular programming interface. In this paper, we will compare the two ways of going about creating the instrument interfaces. We will also present how to integrate the system into an online course distribution environment in keeping with current norms, standards and requirements that have either been adopted or are in the process of being adopted, such as Instructional Management Systems (IMS) and its derivative norms.*

INTRODUCTION

In research projects on remote laboratories, it is becoming more and more common to find acquisition, measurement, visualization and analysis tasks. Technological advances in the design of network interlock systems give rise to a significant and increasing demand for implementation of distributed systems concepts for remote measurement, remote operation or remote control. In the context of a global economy, there is tremendous potential for finding applications for these concepts in many fields, such as health, training, the environment and mining, to name only a few. While from a technological point of view, many aspects of these concepts appear to be familiar to us already, this is not yet the case for task management and organization [1]. Nor is it the case for putting online the adaptive and user-friendly interfaces required by participants in collaborative projects according to the role they are to play during lab sessions conducted in synchronous mode [2]. Through these interfaces, it is possible to provide remote or local access for each participant to the tools and/or equipment that they will be using virtually — alone or in collaboration with others.

This paper will present a few of the experiments that we have conducted to design and put online interfaces through which it is possible to retrieve data and make measurements by accessing remote measuring instruments in a pedagogical context in which a collaborative learning model is implemented.

THE RVLABX ENVIRONMENT

RVLabX (Figure 1) is a virtual laboratory environment that provides a means for participants in a lab session to interact with real instruments via graphical interfaces that reproduce the interfaces of remote apparatus or devices through networks and under the supervision of a trainer and/or an attendant [3]. According to the activity scenario given to learners, the trainer or attendant provides each member of the delocalized group with the appropriate virtual interfaces that will allow them to interact with the system when the time comes. The interfaces are designed such that each participant can play a role during a synchronous session of lab work. The parameters controlled by each participant can thus be managed by means of a session management application that we have developed [2]. The choices of the trainer are received on the participants' side by a component that displays the status of the session, namely, who is present, who sees what and who is controlling what (Figure 2). The application communicates with the component displaying the virtual interfaces of the instruments so that it may display the interface selected by the manager, who may also act as the trainer. At any time during a session, the manager may choose to activate or deactivate the controls of a particular interface.

[1] Hamadou Saliah-Hassane, Télé-Université, LICEF/CIRTA Research Center, 4750, Henri-Julien, Montréal, H2T 3E8 saliah0@licef.teluq.uquebec.ca
[2] Patrick Dumont-Burnett, LICEF/CIRTA Research Center, 4750, Henri-Julien, Montréal, H2T 3E8 pdb1@hotmail.com
[3] Colombiano Kedowide LICEF/CIRTA Research Center, 4750, Henri-Julien, Montréal, H2T 3E8 colombiano.kedowide@hec.ca

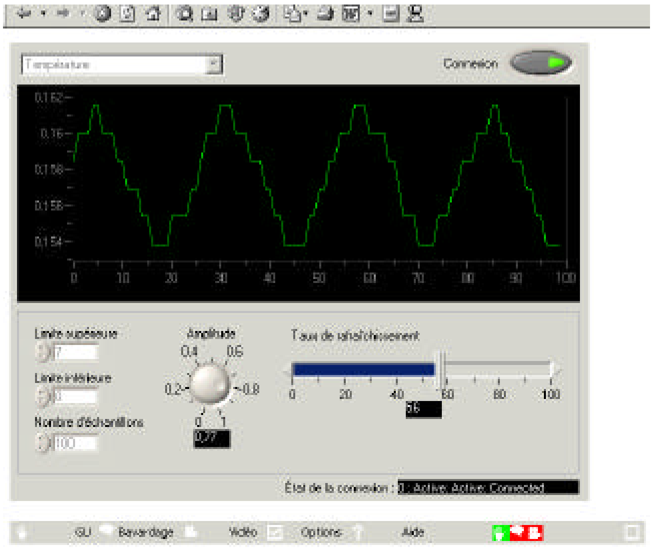

Figure 1 **User Distributed Virtual Instrument Interface [2]**

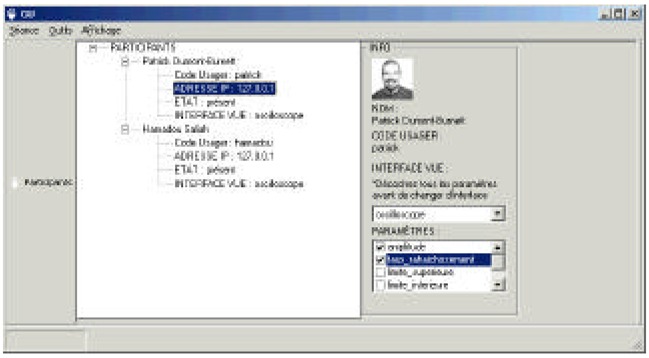

Figure 2 **User Management Application**

OVERVIEW OF TWO CONTROL MODELS DISPLAYING VIRTUAL INSTRUMENT INTERFACES

To be compatible with the synchronous lab-work session management system, the component charged with displaying the virtual instrument interfaces must implement a programming interface that we have defined. It is through this programming interface that a particular interface can be displayed and particular control parameters can be activated or deactivated. We used two models of components displaying virtual instrument interfaces.

The first interface model is an ActiveX control that we have named Polymorphix, embedded in an application able to host ActiveX components, which happens to be the browser Internet Explorer in this particular case. The control displays interfaces whose parameters are defined in an XML file. These interfaces, as we have already mentioned, are created in the interface generation application, which helps the trainer or attendant design the user interface of each participant by dragging and dropping controls—which can be buttons, cursors or graphical visualization windows—onto a control panel. Through properties windows, the parameters of these controls can also be set to adjust their size, determine their scale or link the data source to specific Internet Protocol addresses. The objects thus created, which make up the interfaces and their parameters, are saved in XML files to be read by Polymorphix during the session.

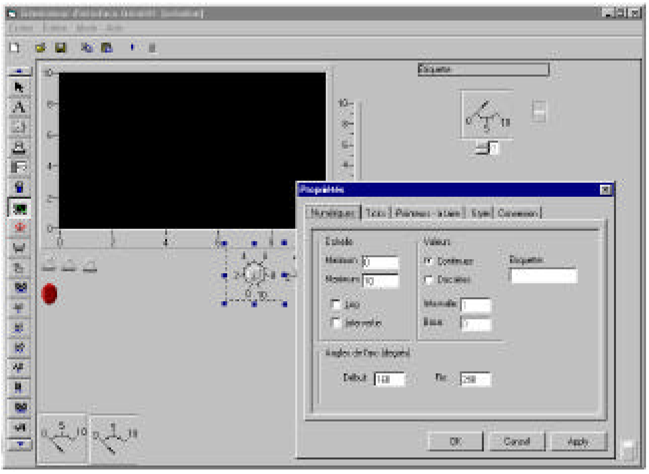

Figure 3 **Graphic User Interface Generator [3]**

The second model involves creating ActiveX controls from a development environment in which it is possible to develop ActiveX components (Microsoft Visual Studio, for example). These controls are embedded in an application able to host ActiveX components (in our case, the control is embedded in Internet Explorer). All of the virtual instrument interfaces needed to conduct an experiment are compiled in a single control. Table 1 presents the pros and cons of each of the models that we have just described.

Table 1: Table for comparison of the two models

XML Interfaces	Compiled Interfaces
A single ActiveX control can display several interfaces. Does not require a recompile.	All interfaces needed for a lab session must be compiled in a single ActiveX control. It is not possible to add interfaces to the control without recompiling.
A single electronic signature is needed for infinite sets of interfaces.	Requires as many electronic signatures as there are interface sets in the same ActiveX control.
Interfaces are programmed using a graphical development environment that is relatively easy to use.	Interfaces must be programmed using a programming language that allows creation of ActiveX

	controls.
The level of sophistication of the interfaces is dependent on the development environment of the XML interfaces. The more sophisticated the application, the more flexible the interfaces can be.	The level of sophistication of the interfaces is limited only by the flexibility of the programming language used and by the developers imagination.
The use of new types of controls for the interfaces requires a recompile of the development environment and of the Polymorphix control. This means that a new electronic signature will be issued for the environment.	

WHY DESIGN VIRTUAL INSTRUMENT INTERFACES AS LEARNING OBJECTS?

In this paper, we consider a learning context in which participants, learners and remote trainers interact during a synchronous lab session by means of distributed interfaces. These interfaces provide access not only to simulations (in some cases), but to real equipment as well. Each interface made available to a participant can therefore represent a local or remote device. The new breed of lab attendant must make a certain number of interfaces available to participants, enabling them to play their roles and must assign privileges, before or after the start of the session, granting total or partial control of shared equipment. To make the remote lab attendants' task easier, we propose to establish a bank of learning objects. This bank of learning objects would constitute the interoperable shared interface resources, meeting the terms of IMS standards [4] and addressing the concerns expressed by Ramsley et al. [5].

The primary characteristics of the IMS standards are the following [6]: 1) *accessibility* is such that it is possible to search for, identify, access and deliver online training content and components in distributed fashion. 2) *Interoperability* allows for use of content and components developed by an organization on a particular platform by other organizations on other platforms. 3) *Reusability* provides for reuse of content and components for differing purposes in differing applications, differing products, in differing contexts and by means of differing modes of access. 4) *Durability* allows content and components to deal with technological changes, without requiring redesign or redevelopment. 4) *Adaptability* provides for custom modulation of content and components.

CONTROLS DISPLAYING VIRTUAL INSTRUMENT INTERFACES AS LEARNING OBJECTS

As both models for controls that we have developed are ActiveX components, besides the interface of compatibility with the management system, they can present the application that hosts them to a programming interface whereby it is possible to set their parameters. For example, in the case of the control of the compiled interfaces, the address of the data source is defined in the Web page where the control is found by means of a property shown by the control. Thus, the virtual instrument interface displayed by the control can be linked to another instrument of the same type but located elsewhere on the network. Other properties could also be defined, providing for greater independence from the operating environment. The control can thus become a scaleable learning object that can be reused in differing contexts, without set outside dependencies. The controls are currently used in a multi-user, supervised context, but it would be quite possible to come up with a scenario for a single, unsupervised user, according to the mechanism for showing the properties.

CONTROLS DISPLAYING THE VIRTUAL INSTRUMENT INTERFACES DISPLAYING STANDARDIZED INTERFACES

The controls displaying virtual instrument interfaces can be designed to be compatible with certain standards, such as those proposed by the Interchangeable Virtual Instrument Foundation (IVI), which defines instruments classes. As these instrument classes present standardized interfaces, the controls displaying virtual instrument interfaces will not only be independent of the physical location of the instrument providing data, they will also be independent of the particularities of this instrument, thus providing for broader reuse of the control. What we suggest, in fact, is that a software standard for measurement instruments be established.

With the development tool *Measurement Studio*™ provided by *National Instruments*™, we are in a position to propose solutions that meet the desirable and required characteristics dictated by current trends in norms and standards in relation to IMS learning objects. Let us recall that these characteristics are accessibility, scalability, reusability, interoperability, adaptability and durability. Measurement Studio provides a complete set of powerful ActiveX controls for Visual Basic, as well as containers for other ActiveX controls and powerful libraries of object-oriented classes designed for creating advanced measurement applications in Visual C++™. There is also the possibility offered by LabWindows/CVI™. If the use of IVI instruments makes it possible to avoid dependency on a particular instrument supplier in a virtual laboratory application, we can, with these drivers, use a high-level application programming interface (API) to communicate with the instruments used.

By calling, for example, a simple function such as "Read waveform", we don't need to know all of the low-level commands to send to our instrument, but we also have the ability to easily interchange instruments provided by different suppliers without having to rewrite or change our code. Change a value in our configuration, and our code will be able to use a driver with any compatible IVI instrument that fits into one of the five instrument classes already adopted, namely Oscilloscopes, Digital Multimeters, Function Generators, Switches, and Power Supplies.

Measurement Studio was designed using the latest standard technologies, such as ActiveX and COM. Other development tools and other equipment suppliers are able to achieve the same objectives. Open source code solutions , on the other hand, could guarantee the interoperability and durability sought by the normalization and standardization process, among other characteristics,.

CONCLUSION

In this paper, we discussed our preoccupation with starting a process to standardize the process of putting online and distributing synchronous remote lab activities. To a great extent, the premises of this paper rest on the research work regarding virtual laboratories that we have been conducting in recent years. We have proposed paths for the construction of distributed interfaces, based on the availability of emerging, powerful tools that meet the requirements of standards on learning objects. The community of trainers and learners could draw benefits from the resource banks adapted to the context of network-based collaborative lab work.

REFERENCES

[1] H. H. Saliah, M. Saad, L. Villardier, B. Assogba, C. Kedowide, T. Wong, " Resource Management Strategies for Remote Virtual Laboratory Experimentation ", "Proceedings of the 2000 Frontier in Education Conference"; Building on a Century of Progress in Engineering Education, Kansas City, October 18-21, 2000.

[2] H. H. Saliah, C. Loizeau, P. Dumont-Burnett, "Design of a Web-Based Virtual Laboratory Instrument Measurement Interface", "2001 International Conference on Engineering Education"; Interfacing the World, Oslo/Burgen, Norway, August 6-12, 2001.

[3] H. H. Saliah, P. Dumont-Burnett, C. Loizeau "RVLabX, a Web-based Interactive Laboratory Environment for Education and Research", "2001 Frontier in Education Conference"; The Future - Impact on Engineering and Science Education, Reno, October 10-13, 2001.

[4] V. Licks, F. Leldens and al., "Learning objects : a model for collaborative content production and a case study", *International conference on engineering education*, Interfacing the World, Oslo/Burgen, Norway, August 6-12, 2001

[5] Ramsley D. J., Hummels D. M., Segee B. E., "A Virtual Instrument Bus Using Network Programming", *IEEE Instrumentation and Measurement Technology Conference*, pp. 694-697, May 19-21, 1997, Ottawa, Canada.

[6] Simard C., "E-Learning Normalization, Trends and Perspectives", *NordSud.org* ., February 2002

ANNEX 1: AN EXAMPLE OF A RESOURCE DESCRIPTION FILE (RDF) BASED ON CANCORE NORM FOR THE OSCILLOSCOPE (FIGURE 1)

```
<?xml version="1.0"?>
<record xmlns="http://www.imsproject.org/metadata/">
  <general>
   <title>
      <langstring lang="en">oscilloscope</langstring>
   </title>
   <language>en</language>
   <description>
      <langstring lang="en">virtual oscilloscope</langstring>
   </description>
  </general>

  <lifecycle>
   <version>
      <langstring>1</langstring>
   </version>
   <contribute>
      <role>Author</role>
      <centity>
                <vcard>
BEGIN: vCard
N: Hamadou Saliah
ORG:LICEF
EMAIL;INTERNET:Hamadou_Saliah@teluq.uquebec.ca
END: vCard
        </vcard>
      </centity>
   </contribute>
  </lifecycle>

  <metametadata>
   <contribute>
      <centity>
               <vcard>
BEGIN: vCard
N: Hamadou Saliah
ORG:LICEF
EMAIL;INTERNET:Hamadou_Saliah@teluq.uquebec.ca
END: vCard
                           </vcard>
      </centity>
      <date>
         <datetime>2002-06-07</datetime>
      </date>
    </contribute>
    <metadatascheme>CanCore1.0</metadatascheme>
    <language>en-CA</language>
  </metametadata>

  <technical>
    <format>
       <langstring lang="en">application/x-activex</langstring>
    </format>
    <size>2000000</size>
    <location
type="URI">http://telelab.licef.teluq.uquebec.ca/oscilloscope</location>
  </technical>
  <educational>
   <learningresourcetype>
      <langstring lang="en">InteractiveResource</langstring>
   </learningresourcetype>
   <intendedenduserrole>
      <langstring lang="en">learner</langstring>
   </intendedenduserrole>
   <learningcontext>
        <langstring lang="en">Higher Education</langstring>
   </learningcontext>
   <typicalagerange>
      <langstring>Adult</langstring>
   </typicalagerange>
  </educational>

  <classification>
     <purpose>
                <langstring lang="en">discipline</langstring>
             </purpose>
             <taxonpath>
                <source>AlbertaLearningUniversity</source>
                          <taxon>
                             <entry>engineering</entry>
                          </taxon>
             </taxonpath>
  </classification>
</record>
```