

REXIB: Remote Experiments Interface Builder

Jorge R. Alves¹, Gustavo R. Alves², Paulo A. Ferreira³ and Jose M. Ferreira¹

¹ FEUP / DEEC, Porto, Portugal

² ISEP / DEE, Porto, Portugal

³ ISEP / DEI, Porto, Portugal

Abstract— Remote Experimentation is an educational resource that allows teachers to strengthen the practical contents of science & engineering courses. However, building up the interfaces to remote experiments is not a trivial task. Although teachers normally master the practical contents addressed by a particular remote experiment they usually lack the programming skills required to quickly build up the corresponding web interface. This paper describes the automatic generation of experiment interfaces through a web-accessible Java application. The application displays a list of existent modules and once the requested modules have been selected, it generates the code that enables the browser to display the experiment interface. The tools' main advantage is enabling non-tech teachers to create their own remote experiments.

Index Terms— educational aids, remote experimentation, interface builder

I. INTRODUCTION

A review of several interfaces to remote experiments reveals the following modules or components to be usually present: (1) measuring instruments; (2) a dedicated experimental apparatus with control mechanisms; (3) audio/visual feedback from the remote lab; and, (4) one or more communication channels for group work or tutor assistance [1, 2, 3]. More sophisticated remote experiments sometimes include a combination of simulations modules for creating mixed-reality environments. In general, these components are either Java Applets, Virtual Instruments (VIs) built with LabView, Flash applications, or some other form of interface building blocks developed with software packages that may require downloading a plug-in. This scenario is thought to be constraining for non-tech teachers willing to develop interfaces for remote experiments supporting courses on science & engineering areas. Even if the teacher is familiar with Learning Management Systems (LMS), such as Moodle [4], for developing and making available the theoretical contents, to build up the interface to a remote experiment and link it with the LMS, eventually through a scheduling / booking system, are not trivial tasks. A possible solution is to develop an automatic interface builder.

II. ANALYSIS OF REQUIREMENTS

Such a solution should consider two perspectives: the teacher's and the students' perspectives. The first should be as simple as possible while at the same time allowing the teacher to select among a list of available modules for

building up a particular remote experiment interface. At the same time all detailed aspects of combining the generated interface with a booking system associated with a particular LMS should be hidden from the user, i.e. they should be handled by the automatic interface builder. These two requirements lead to a first draft of the teacher's perspective, as depicted by figure 1.

(Remote experiment interface)	<input checked="" type="checkbox"/> Oscilloscope
	<input type="checkbox"/> Waveform generator
	<input type="checkbox"/> Communication (Skype)
	<input checked="" type="checkbox"/> Communication (MSM)
	<input checked="" type="checkbox"/> Webcam n.º 1
	<input checked="" type="checkbox"/> Webcam n.º 2
	<input checked="" type="checkbox"/> Webcam n.º 3
	<input type="checkbox"/> ...

Figure 1. Initial spec of the teacher's perspective when building up the remote experiment interface

After selecting the modules required for building up the remote experiment interface, the application should produce what is here identified as the student's perspective. Figure 2 illustrates a first draft of this student's perspective, i.e. the remote experiment interface as seen from the client machine web browser, with all the modules indicated by the teacher.

(Control mechanisms: sharing, booking, etc.)	(Chat)
Webcam	Webcam
Webcam	(Chat text input)
(Interface with remote equipment / devices)	

Figure 2. Initial spec of the student's perspective

An additional requirement (not implemented in the presented solution) was the possibility to track and record the student activity when using the remote experiment interface. This should return some meaningful data for assessment purposes.

III. A REMOTE EXPERIMENTS INTERFACE BUILDER

REXIB (Remote EXperiments Interface Builder) is a Java-based application that facilitates the development of experiment interfaces. The initial teacher's perspective

depicted in figure 1 evolved to the application interface illustrated in figure 3. It accepts a list of modules that are kept in a library (in one or several machines) and that can be customized through a simple text file. In its present version, REXIB allows the teacher to select LabView VIs, webcam channels (unidirectional for lab feedback, or bidirectional for collaborative purposes), and Virtual Network Computing (VNC) links.

Interface Builder

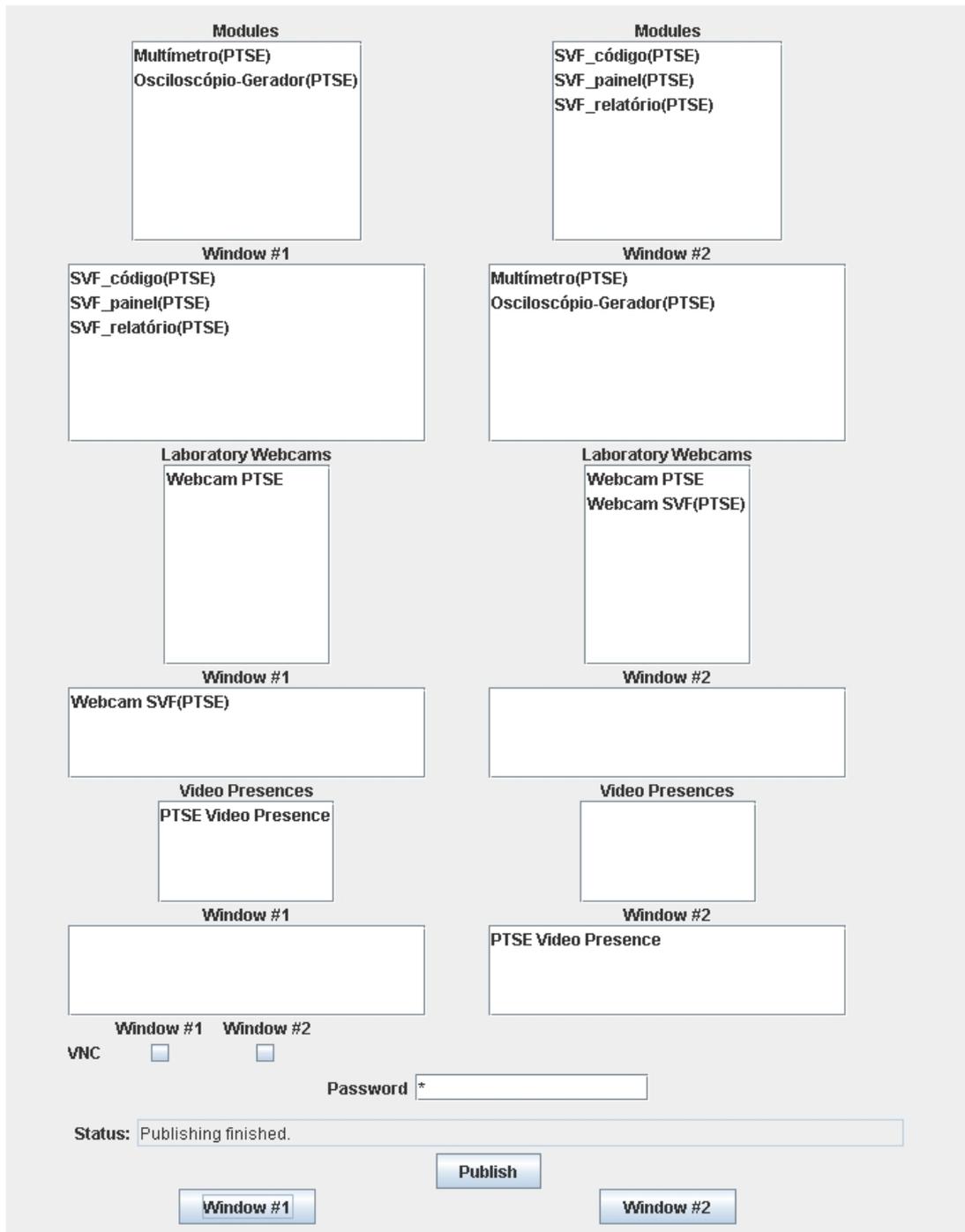


Figure 3. Teacher's perspective from REXIB

These modules can be placed in one or two windows, according to the user selection. Depending on the number and size of the individual modules, a sorting algorithm (embedded within the interface builder) generates a list of best fitting options, which can be individually viewed by the user. These options correspond to the several possible alternatives for the remote experiment interface. The user may choose from a 1st set of cameras for videoconferencing and from a 2nd set for lab feedback, in each case being also possible to place them in window 1 or 2. The possible configurations regarding the VI's placement heavily depend on the number and size of each module. REXIB input information includes: VI description (URL, height, width, name, (optional) Booking system); password specification (for security reasons, as this application is also remotely accessible); target information (where to store the automatically

generated code, and the Internet address); videoconferencing cams (URL, height, width, name, (optional) booking system); labfeedback cams (URL, height, width, name, (optional) booking system), and VNC data (computer id, port id). The interface builder contains buttons for publishing and viewing the windows generated. Figure 4 and figure 5 correspond to two windows generated by REXIB after the teacher selected a particular configuration. In this case, one of the windows (figure 4) contains three VIs and one webcam for lab feedback, and the other window (figure 5) contains four video conference channels and two VIs. After clicking on button "Window #1" (or #2), the URL of the corresponding window will be visible in the address bar of the browser. The application has a built-in mechanism that also allows passing this information to a booking system developed for Moodle [5].

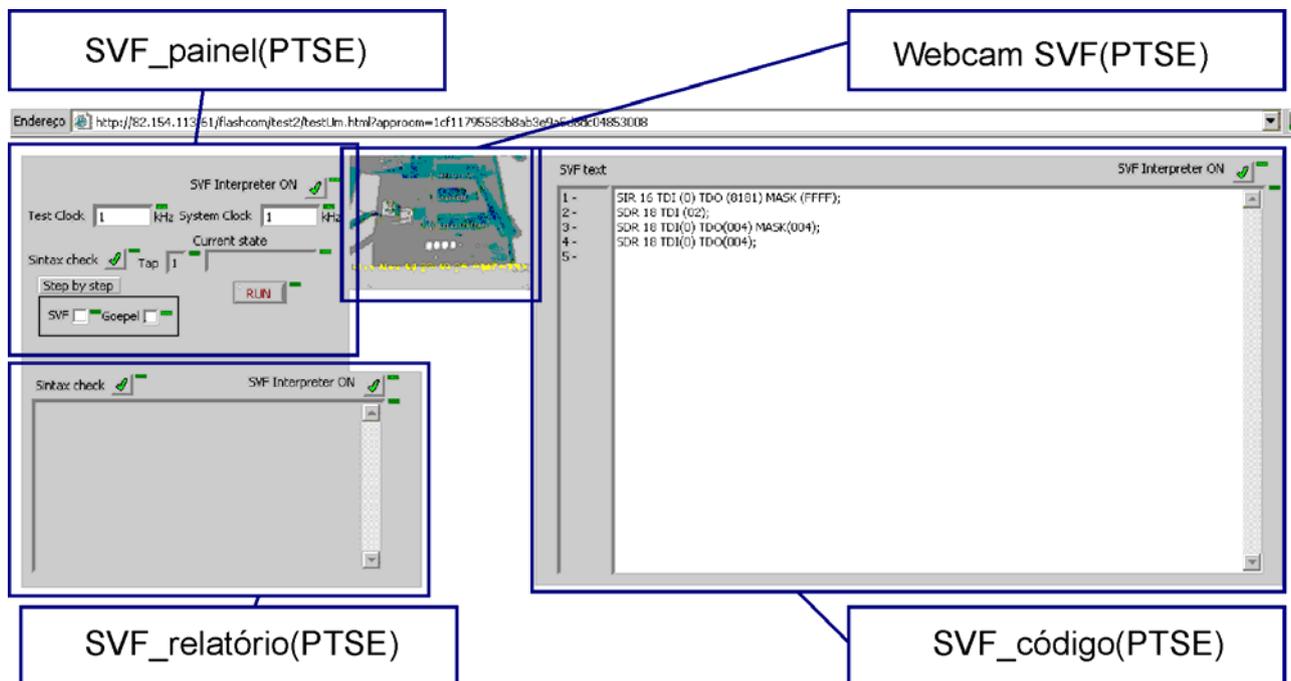


Figure 4. Students' perspective: window 1

IV. CONCLUSION AND FUTURE WORK

The main advantage of the developed solution is to enable non-tech teachers to create their own remote experiments. It requires a low level of understanding from the teacher perspective on how the individual interface building blocks are built and on how they can be arranged to create the interface for accessing a particular remote experiment. Another advantage is the possibility to allow students using REXIB. In this way, the teacher may (through a careful selection of the list of available modules) overcome a limitation usually associated with remote experimentation [6], i.e. allow the students to select the lab equipment required to conduct a certain experiment, and later on assess the students' options.

Regarding future work, it is further necessary to define a protocol between LabView applications, developed and hosted at different Institutions, which

allows their automatic integration into a single web page. There are similar protocols in RPC (Remote Procedure Calls), Java RMI (Remote Method Invocation), CORBA and Web Services. Such a protocol would allow interconnecting measuring devices, located at a given Institution, with data analysis applications or human-computer interface modules, located at other Institutions. This possibility would undoubtedly increase the cooperation opportunities among different Institutions, some more focused on real equipment and others more focused on mathematical applications for data analysis.

ACKNOWLEDGMENT

The authors acknowledge the contribution and suggestions made by Antonio Cardoso from the Faculty of Engineering, University of Porto, in what concerns to linking REXIB with the Moodle booking system.

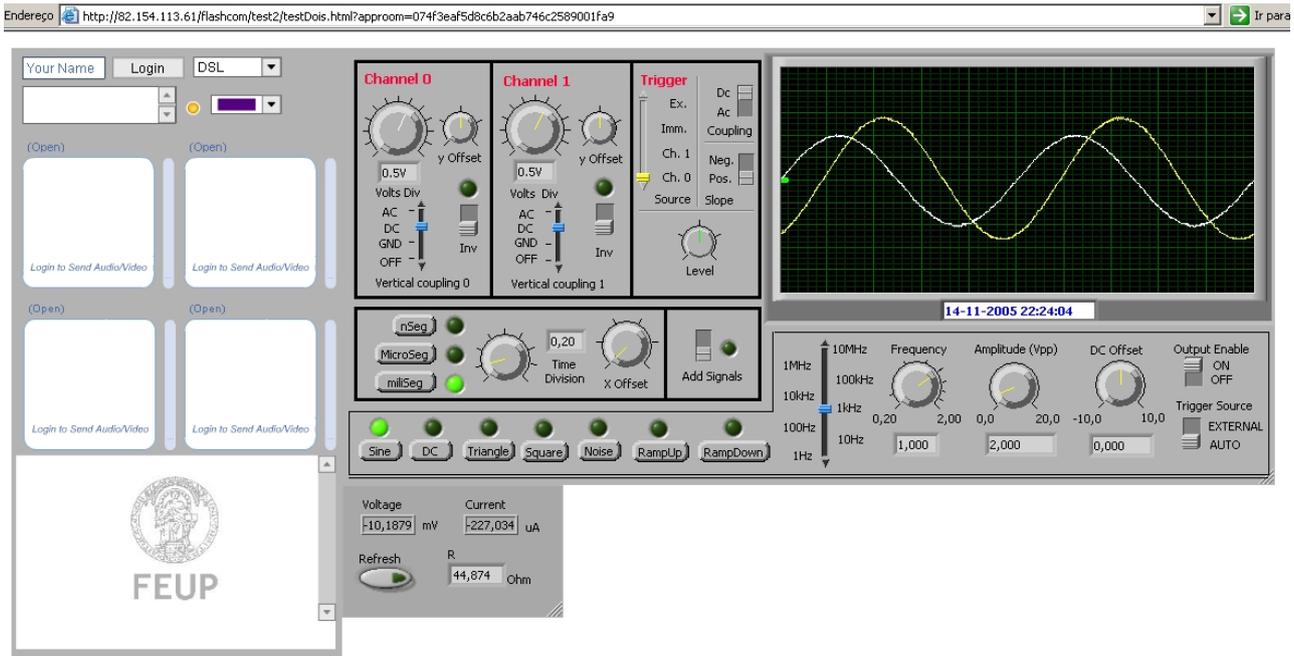


Figure 5. Students' perspective: window 2

REFERENCES

- [1] Ko, C. C., et al. (2001). "A web-based virtual laboratory on a frequency modulation experiment". In: IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, Volume 31, No. 3, pp. 295-303.
- [2] Johnston, Murray V., et al. (2001). "Remote Experimentation over the Net: Our First Year with MALDI". In: Analytical Chemistry, bi-monthly Journal of the American Chemical Society, Volume 73, No. 15, pp 440 A-445 A.
- [3] Müeller, Dieter and Ferreira, José (2003). MARVEL: "A mixed-reality learning environment for vocational training in mechatronics". In: Proceedings of the Technology Enhanced Learning International Conference, Milan, Italy.
- [4] Moodle, <http://moodle.com/> (2006)
- [5] Ferreira, José and Cardoso, António. "A Moodle extension to book online labs." In: International Journal of Online Engineering (iJOE) [Online], 1.2 8 Nov 2005 Available: <http://www.i-joe.org/ojs/viewarticle.php?id=21>
- [6] Zubia, J. Garcia and Moral, A. d. (2005). "Suitability and Implementation of a WebLab in Engineering." In: Proceedings of the 10th IEEE International Conference on Emerging Technologies and Factory Automation, Catania, Italy.

AUTHORS

Jorge R. Alves is with the Electrical and Computer Engineering Department, Faculty of Engineering, University of Porto, Portugal (e-mail: jorge.alves@fe.up.pt).

Gustavo R. Alves is with the Electrical Engineering Department, School of Engineering, Polytechnic Institute of Porto, Portugal (e-mail: gca@isep.ipp.pt).

Paulo A. Ferreira is with the Informatics and Computer Engineering Department, School of Engineering, Polytechnic Institute of Porto, Portugal (e-mail: paf@dei.isep.ipp.pt).

Jose M. Ferreira is with the Electrical and Computer Engineering Department, Faculty of Engineering, University of Porto, Portugal (e-mail: jmf@fe.up.pt).

Manuscript received July 11, 2006.