Self Growing Remote Controlled Laboratory

P. Cotfas, D. Ursutiu and C. Samoila
“Transilvania” University of Brasov - Romania

***Abstract* — In the last few years, focused on the collaboration with a team from Carinthia Tech Institute Villach -Austria, we have tried to develop a remote controlled laboratory in the Electronic Engineering field at the “Transilvania” University of Brasov. The main idea of this paper is to present the improvement of our LabVIEW Server, access control and service management. The work was concentrated to develop an interface for our server which provides the possibility to simultaneously connect for many users to many applications. A new task for our work was to develop possibility for clients to make a scheduler for access the applications. Another direction of research was to create a possibility to adding on-line at our laboratory, any new developed application. If it is started, the server detects it and shares it for the client.**

***Index Terms* — Remote control, self-growing, data socket, virtual laboratory.**

I. Introduction

In the last years, the Open and Distance Learning has known a very accentuated development, being taking over and recognized by more and more counties and universities as an alternative way of learning and perfecting. There are companies that have well organized structures of perfecting based on the ODL system in their field of activity, such as CISCO, Microsoft etc. These companies give the possibility of perfecting by offering the study materials on-line, such as: courses and seminaries, tests for verifying the attained level, exams for courses graduation etc. Many universities have developed departments dedicated to ODL. Still, all of these studies are oriented towards the fields that do not require laboratories that might need the maneuverability of certain devices or studies of real phenomena. The necessity of a laboratory brings big organizing problems: the space, available qualified personnel, high costs etc. In certain fields of ODL where the laboratories are necessary, has been taken over the idea of joining the ODL with the standard learning by creating some local academies where the students could achieve the necessary preparation in laboratories. So, in applied fields such as the engineering one, the accomplishing of the ODL system would require some compromising that in the most of the cases would lead to the fact of not taking into consideration this modality of study.

In order to avoid these local laboratories, there is an ascendant flow of creating the so-called virtual laboratories, though them the troubles of the ODL in the applied fields are overcame.

In this article we wish to present a simple possibility of creating real laboratories remote controlled that would allow the continuing growing number of new applications. A team of Carinthia Tech Institute Villach - Austria, stated this direction of research and this has been developed in collaboration with our team from CVTC, Transylvania University of Brasov.

Together with Carinthia Tech Institute and other universities from Europe it was started the first European Master in the field of Remote Engineering (Project “MARE”). In the future, our applications will be used in a virtual laboratory for the students of this master.

The creation of laboratories like these can be oriented on two directions of approaching:

- The creation of some applications that allow the study of some physical, chemical etc. phenomena or of the behavior of the investigated systems, by selecting the available parameters without having the possibility of step by step controlling these devices in the system. This approach allows the understanding of the phenomenon and not of the maneuverability of the devices in the system;
- The creation of some interfaces for effective step by step control of the devices in the system thus allowing the learning of physic devices maneuverability.

By joining the two directions of approaching it is allowed the creation of some complex laboratories of study.

By making these laboratories functional, the area of covering of the ODL system could be extended. A very important sector is the engineering one. The chosen applications for our laboratory are taking over the field of Electronic Engineering.

II. RELBV Data Socket Sever

RELBV Data Socket Sever is a LabVIEW application that plays the part of a server. The purpose of the application is to allow the connecting of several clients to many applications spread on different workstations. This concept assumes that at one moment of time one client can control one or more applications and another client can control other applications.

The used communication protocol is data socket transfer protocol (dstp). In order that this application should function, the data socket server given by the National Instruments in the LabVIEW package having the afferent rights for writing and reading, must be started on a station that is visible on the Internet.

Due to the fact that the communication with this server is being done on the 3015 port, this port must be opened on that particular station. Using this technology we improved the security for our applications, because we have only the NI-Data Socket Server in a demilitarized zone and the rest of our applications: the server application and the applications can be wherever on the network without the knowledge of the client.

The functioning concept is described in the below image:

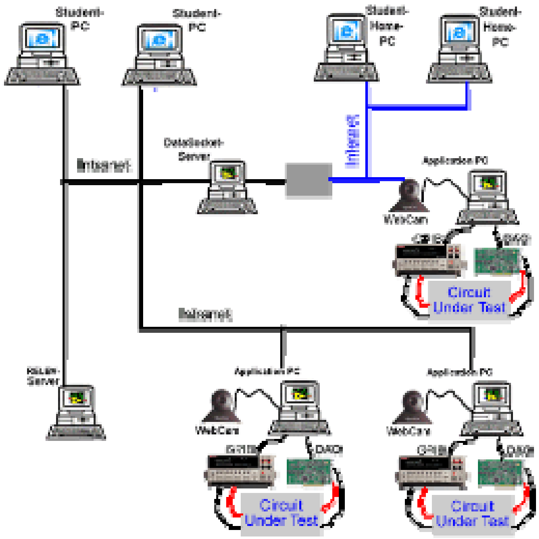


Figure 1. The distribution in the network of the system's components

As you can see from Fig.1, the data socket applications can be spread anywhere in the computer network (both in the Intranet and Internet). The "Data Socket Server" is doing the connection between them.

The problem might appear only for publishing the feedback image by using a web camera. In order to do this there can be used some software that would write the image taken by the web camera on the web server (the server that allows the access to the virtual laboratory site). Using a java script, this image can be published in the web with a settled scan rate (no less than 0.5 s that depends on the speed of the connection with the client). The stations in the Internet can publish by themselves this image even using some streaming software.

The software required in order that this project should function is the graphical programming language LabVIEW, Component Works and Data Socket Server, given by National Instruments, Internet Explorer and the Microsoft-ActiveX Control technology.

A. The Functioning of RELBV Data Socket Server Application

RELBV_DSS has as purpose the clients' login, verification of sever-client connections, client's access to the application, testing of applications' status (functional or not), and client's scheduled programming to different applications.

In order to accomplish these functions, the RELBV_DSS application memorizes the information in a database called "Baza_de_Date_Users".

In this database are registered the accounts and passwords of these clients, the data regarding the connecting of a client (history: date and hour of connecting and disconnecting), the state of each connected client: new entered, busy, busy-scheduler (status dedicated to the clients for making the schedule), or in the state of exit (Time Out or Lost Connection). In this database are also registered the status of the applications: available (0), busy (1), reset (2) or unavailable (3).

The used database has been created in Microsoft Access. The communication from LabVIEW with the database is using the Data Connectivity Tools Add-on.

The application is being built on 4 main modules (Fig. 3) ServerUser.vi, Nucleu2.vi, NucleuAccessType.vi and ServerApplications.vi.

ServerUser1.vi (1) is waiting the request for access from a client, verifies and allows the right of access for this client (in the case of access, the client is being registered in the database, receiving the status "enter" and the information about the name of the applications registered in the database).

Nucleu2.vi (2) and NucleuAccessType.vi are applications that play the part of starting a new instance of the InstanceUser.vi and Instance UserAT.vi applications for every new user connected to the server. InstanceUser.vi controls the state of each user logged to the applications server, while InstanceUserAT.vi controls the state of each user connected to the scheduler area. So, for each user a new instance is created, that runs independent of the other instances of the other users.

At the starting of the instances for the two applications, the users are being put in the "Busy" and "BusyS" state.

Here, the existence of the connection with the clients and the status of the access time are verified.

In the case of lost connection, the user is being deleted from the list of logged uses after 5 verifications of the connection.

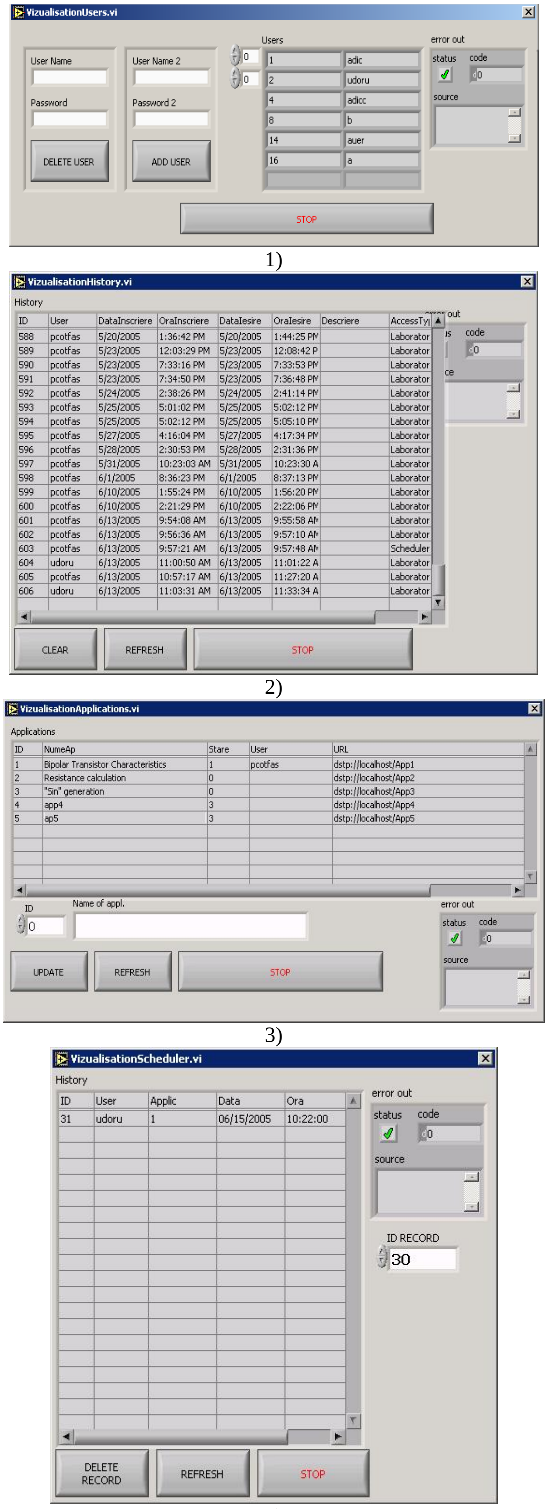
All of the clients receive a settled access time, at whose expiration they shall be automatically disconnected. From here it is also being sent to each client the information about the applications' status (available or not).

ServerApplications.vi. is starting one instance of the ApplicationControl.vi application for each application that is registered in the database. ApplicationControl.vi makes the connection between the server and the applications registered in the database. ApplicationControl.vi is checking the applications' status (running or not) and it writes it in the database.

If a client is connecting to one of the applications, ApplicationControl.vi puts this application as being "Busy" (and no other client cannot connect to this application) and warns it about the existence of a client that is connecting to it and so the application shall enter in the control behavior.

Many facilities for controlling the application have been introduced for the management of the server (Fig.2), such as:

1. The module for editing clients' accounts. The allowed operations are the deleting of a client or introduction of a new client;
2. The module for visualizing the history of server's activity (connected clients, date, hour of the connection, the activities accomplished in the laboratory);
3. The module for editing and visualization of applications' status (the name of the positioning application etc.);
4. The module for scheduler that allows the visualization and deletion of clients' schedule.



VizualisationUsers.vi

User Name

Password

User Name 2

Password 2

Users

1	adic
2	udoru
4	adicc
8	b
14	auer
16	a

error out: status ✓ code 0; source

1)

VizualisationHistory.vi

History

ID	User	DataInscriere	OraInscriere	DataIesire	OraIesire	Descriere	AccessTy
588	pcotfas	5/20/2005	1:36:42 PM	5/20/2005	1:44:25 PM		Laborator
589	pcotfas	5/23/2005	12:03:29 PM	5/23/2005	12:08:42 P		Laborator
590	pcotfas	5/23/2005	7:33:16 PM	5/23/2005	7:33:53 PM		Laborator
591	pcotfas	5/23/2005	7:34:50 PM	5/23/2005	7:36:48 PM		Laborator
592	pcotfas	5/24/2005	2:38:26 PM	5/24/2005	2:41:14 PM		Laborator
593	pcotfas	5/25/2005	5:01:02 PM	5/25/2005	5:02:12 PM		Laborator
594	pcotfas	5/25/2005	5:02:12 PM	5/25/2005	5:05:10 PM		Laborator
595	pcotfas	5/27/2005	4:16:04 PM	5/27/2005	4:17:34 PM		Laborator
596	pcotfas	5/28/2005	2:30:53 PM	5/28/2005	2:31:36 PM		Laborator
597	pcotfas	5/31/2005	10:23:03 AM	5/31/2005	10:23:30 A		Laborator
598	pcotfas	6/1/2005	8:36:23 PM	6/1/2005	8:37:13 PM		Laborator
599	pcotfas	6/10/2005	1:55:24 PM	6/10/2005	1:56:20 PM		Laborator
600	pcotfas	6/10/2005	2:21:29 PM	6/10/2005	2:22:06 PM		Laborator
601	pcotfas	6/13/2005	9:54:08 AM	6/13/2005	9:55:58 AM		Laborator
602	pcotfas	6/13/2005	9:56:36 AM	6/13/2005	9:57:10 AM		Laborator
603	pcotfas	6/13/2005	9:57:21 AM	6/13/2005	9:57:48 AM		Scheduler
604	udoru	6/13/2005	11:00:50 AM	6/13/2005	11:01:22 A		Laborator
605	pcotfas	6/13/2005	10:57:17 AM	6/13/2005	11:27:20 A		Laborator
606	udoru	6/13/2005	11:03:31 AM	6/13/2005	11:33:34 A		Laborator

error out: code 0

2)

VizualisationApplications.vi

Applications

ID	NumeAp	Stare	User	URL
1	Bipolar Transistor Characteristics	1	pcotfas	dstp://localhost/App1
2	Resistance calculation	0		dstp://localhost/App2
3	"Sin" generation	0		dstp://localhost/App3
4	app4	3		dstp://localhost/App4
5	ap5	3		dstp://localhost/App5

ID 0

Name of appl.

error out: status ✓ code 0; source

3)

VizualisationScheduler.vi

History

ID	User	Applic	Data	Ora
31	udoru	1	06/15/2005	10:22:00

error out: status ✓ code 0; source

ID RECORD 30

4)

Figure 2. Interfaces for Server Management

B. *The Interface of RELBV Data Socket Server*

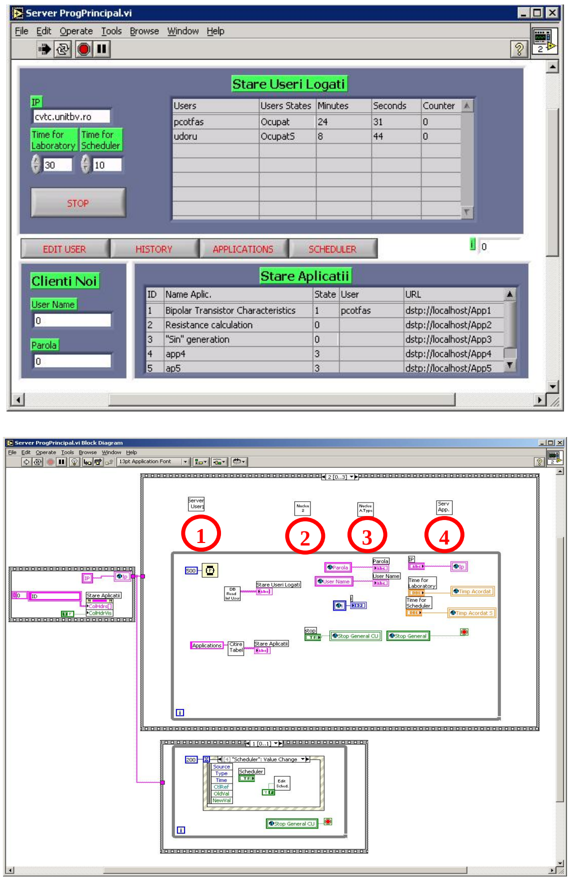
The interface of the application allows setting and visualizing of the next parameters:

Controls:

1. IP (URL address or the IP of the station on which runs the Data Socket Server);
2. Allowed Time – the value of the time allowed to the clients for a work session (for example: 30 min for the lab, 10 min for scheduler). The time can be ulterior changed.
3. Edit User – calls the application for editing the clients' accounts;
4. History – calls the application for visualizing the history;
5. Applications – allows editing the applications;
6. Scheduler – allows editing and visualizing the schedule for clients;
7. Stop – Stops the application.

Indicators:

1. Logged User Status – allows the visualization of the working clients:
   a. The name of the clients;
   b. The remained time for working: minutes and seconds;
   c. The connection counter (>5 the client will be automatically disconnected)
2. New clients - allows the visualization of the clients that try to connect (the information remain active only during the communication with the potential client);
3. Applications' status – presents the number, the name and the status of the application (0-available, 1-busy, 2-reset, 3-not connected to the server). If a client has been connected to an application, his name will also appear.



Server ProgPrincipal.vi

Stare Useri Logati

IP: cvtc.unitbv.ro

Time for Laboratory: 30

Time for Scheduler: 10

Users	Users States	Minutes	Seconds	Counter
pcotfas	Ocupat	24	31	0
udoru	OcupatS	8	44	0

Clienti Noi

User Name: 0

Parola: 0

Stare Aplicatii

ID	Name Aplic.	State	User	URL
1	Bipolar Transistor Characteristics	1	pcotfas	dstp://localhost/App1
2	Resistance calculation	0		dstp://localhost/App2
3	"Sin" generation	0		dstp://localhost/App3
4	app4	3		dstp://localhost/App4
5	ap5	3		dstp://localhost/App5

Figure 3. User Interface and Diagram for RELBV_DSS Application

C. *Remote Controlled Applications*

The application from our virtual laboratory can be placed anywhere in the Intranet or even in the Internet. This possibility enhances the number of applications which can be included in our virtual laboratory. If a new application is started and if the application is in the database, the server detects it and shares it for the client. Due to this way of functioning, our virtual laboratory is self-growing, meaning that any new created and running application is being included in the laboratory.

The server for applications verifies if the application is running or not, so there was necessary the implementation of a new method of answering for the server. There is the possibility that the Data Socket Server is shut down, so the connection will be lost and the application will need to be restarted.

In order to avoid such a situation, a procedure for testing the existence of the connection to the server has been included, or else, at 5 unsuccessful repeated attempts the application is automatically restarted. The restart procedure is being executed until the restoring of the connection.

The interface of the applications has been divided into two areas:

- The area dedicated to the controls and indicators used for visualizing the connections;
- The area dedicated to the studied phenomenon (see fig. 4).

D. *ActiveX*

The remote control of the applications is made through web. For each application has been made an ActiveX that has been published in an html page. The ActiveX were made in Microsoft Studio 6, Visual Basic. These allow the remote interaction via Internet with the applications through data transfer assured by the Data Socket Server.

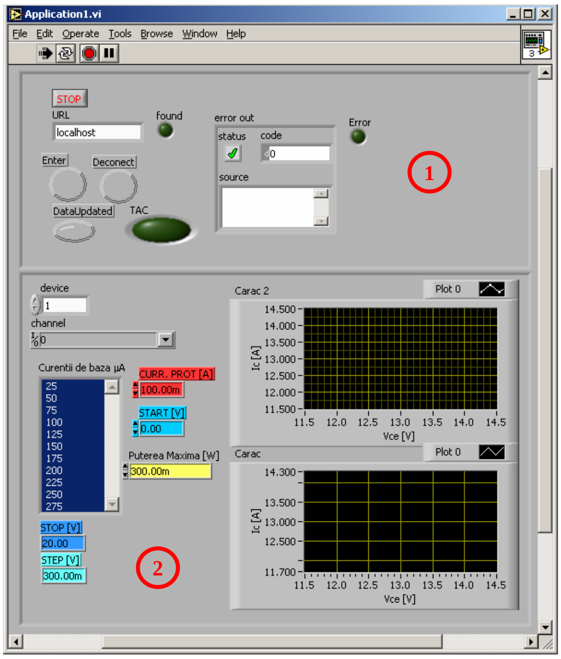


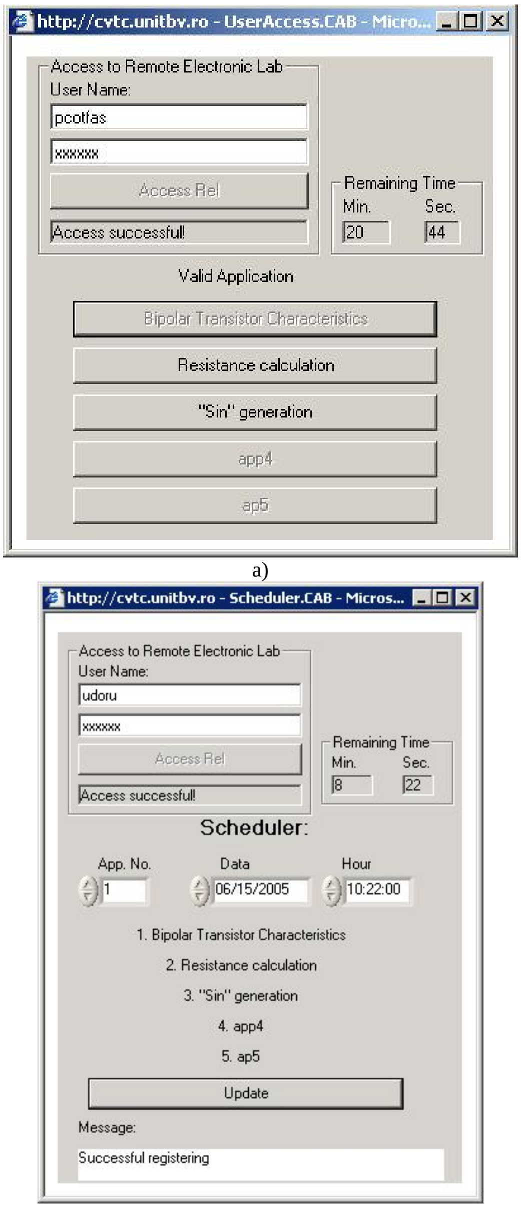
Figure 5. The interface for Application 1
(The study of the collector characteristics of a bipolar transistor)

The first ActiveX allows the registering of the client and if successful, allows the connection to the available applications. Each ActiveX is opened in a new window.

Page accessing can be done only with Internet Explorer, whose properties must be set so that it allows the installing and running of ActiveX. In order to do that, it is necessary that at installing, the client should have administrator rights.

Another thing that was achieved in our LabVIEW Server is the development of a possibility for clients to make a schedule for accessing the applications (see Fig. 5.b). In this way, the client can be assured that he will have a possibility to connect to the applications that he is interested in.

In Fig. 6 it's shown the ActiveX for one of our applications which have about same structure like the LabVIEW application (Fig.4).


http://cvtc.unitbv.ro - UserAccess.CAB - Micro...
Access to Remote Electronic Lab
Access Rel
Remaining Time
Valid Application
Bipolar Transistor Characteristics
Resistance calculation
"Sin" generation
app4
ap5
http://cvtc.unitbv.ro - Scheduler.CAB - Micros...
Access to Remote Electronic Lab
Access Rel
Remaining Time
Scheduler:
Update

User Name: pcotfas
xxxxxx
Access successfull
Min. 20 Sec. 44

a)

User Name: udoru
xxxxxx
Access successfull
Min. 8 Sec. 22

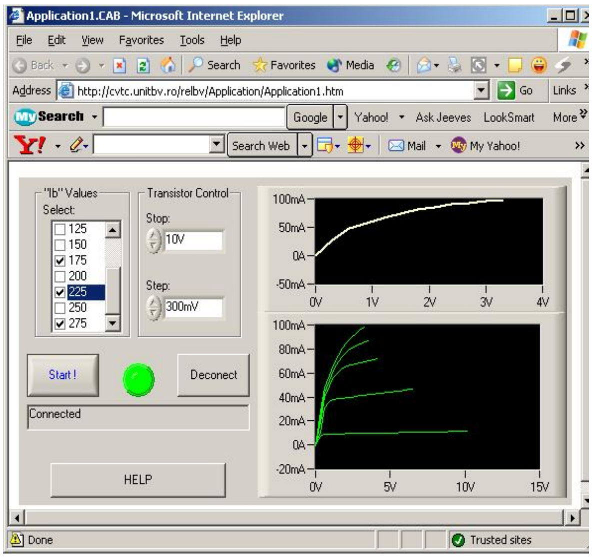
App. No. 1
Data 06/15/2005
Hour 10:22:00

1. Bipolar Transistor Characteristics
2. Resistance calculation
3. "Sin" generation
4. app4
5. ap5

Message:
Successful registering

b)

Figure 4. ActiveX for login for access to lab and for schedule



Application1.CAB - Microsoft Internet Explorer

Address http://cvtc.unitbv.ro/relbv/Application/Application1.htm

"Ib" Values
Select:
□ 125
□ 150
☑ 175
□ 200
☑ 225
□ 250
☑ 275

Transistor Control
Stop: 10V
Step: 300mV

Connected

Done

Figure 6. The ActiveX for Application 1

III. CONCLUSIONS

The creation of some virtual laboratories that allow accomplishing real experiments remote controlled allow the growing of the covering area for ODL in the field of Engineering and of Applied Sciences. In addition to this, there could be made shared experiments in which researchers situated in different locations can share the data and the results.

The partnership between different institutions that participate with this kind of virtual laboratories could lead to an increased quality in the teaching - learning process by using the different kind of devices, instruments and experiments that otherwise would be unavailable. So, instead of the existence of an identical system in every institution, each participant will participate with different systems and so, a wider area of equipments and experiments studied by the students could be covered. This approach could lead to a very big reduction of costs.

Even at a first look the equipments companies might seem disadvantaged, by using a larger variety of equipments in the process of leaning and researching each student and researcher will be better informed over the existence and correct usage, where it is supposed to, of equipments, and so the companies can have a better entrance on their afferent market.

The using of the ActiveX technology for remote control, via web, also brings some trouble due to dependence of the used platform. In order to overcome this trouble, in the future it will be attempted to implement some technologies that are independent of the used platform (for example: using techniques based on Java).

The creation of some self-growing laboratories, in which the number of applications can grow together with their accomplishing leads to a flexible for administrating structure and by using the databases in controlling the laboratory, its management performances can grow.

REFERENCES

[1] Auer, M.E.; Gallent, W. "The "Remote Electronic Lab" as a Part of the Telelearning Concept at the Carinthia Tech Institute", *Proceedings of the ICL2000*, Villach/Austria, 28./29.09.2000

[2] Michael E. Auer "Virtual Lab versus Remote Lab" *20^{TH} World Conference on Open Lerning And Distance Education*, Dusseldorf, Germany, 01-05 April 2001

[3] P. Cotfas, D. Ursutiu, C. Samoila "Using LabVIEW in Computer Based Learning", *Interactive Computer aided Learning Tools and Applications* Ed. M.Auer and U. Ressler, ICL99 Workshop;

[4] P. Cotfas, D. Ursutiu, C. Samoila "Creating a Virtual Lab using LabVIEW" *International Conf. TICE 2000*, Troyes, France;

[5] P. Cotfas, D. Ursutiu, C. Samoila "Virtual Laboratory and Virtual Instrumentation" ,*Internet as a Vehicle for Teaching*, Ed. Susan English, Mihai Jalobeanu, Nicolaie Nistor, Romanian Internet Learning Workshop "RILW2001", august 11-20 2001;

[6] D. Ursutiu "Initiere in LabVIEW Programarea grafica in fizica si electronica", *Editura Lux Libris*, Brasov, 2001;

[7] "LabVIEW User Manual", National Instruments, USA, 1999

[8] J.Travis, "Internet Applications in LabVIEW", *Prentice Hall*, ISBN 0-13-014144-5, 2000

[9] C. Sapijaszko, G. Sapijaszko, "An Innovative Electronics Laboratory System for On Campus and Distance Learning Applications", *Proceedings of the 2004 American Society for Eng.Education Annual Conference & Exposition*, 2004

AUTHORS

P. Cotfas, Transilvania University of Brasov, Department of Physics, 1 Colina Universitatii Brasov (e-mail: pcotfas@unitbv.ro).

D. Ursutiu, Transilvania University of Brasov, Department of Physics, 1 Colina Universitatii Brasov (e-mail: udoru@unitbv.ro).

C. Samoila, Transilvania University of Brasov, Department of Science of Materials, 1 Colina Universitatii Brasov, (e-mail: csam@unitbv.ro).