

Remote experiments using Java: Implementations in the Virtual Electro Lab project

Adrian Pelcz¹, Francisc Sisak¹, Sorin Moraru¹, Silviu Leahu¹, Ionuț Diaconu¹

¹ Faculty of Electrical Engineering and Computer Science,
"Transilvania" University of Brasov, Politehnicii street no.1, Brasov, Romania

Abstract. This paper treats the e-learning domain from the virtual and remote laboratories point of view. These laboratories have Java applets as client side application. The paper presents the importance of offering to students the possibility to do laboratory tests with real data. Although a remote experiment can not provide the student with the real feedback of the experiment, by using real data and a good representation of the laboratory equipment, we can get closer to the real feedback. The article describes the facilities offered by the Java language in making user interfaces for the remote and virtual laboratories. At the same time, we present three implementations in the pilot project "Virtual ElectroLab" of the "Leonardo da Vinci" program that have user interfaces based on the Java Applet technology, for accessing the laboratory resources via Internet.1 Introduction

1 Introduction

Meeting the growing need to have a stronger contact with the classroom and the laboratory, come the virtual classrooms - managed by teachers, the forums, the online testing methods for acquired knowledge and many other tools. The online laboratory and practical work part were less developed, disadvantaging the distance learners.

In this way, the e-learning trend is guided by the emergence of more and more e-learning centers, which offer on-line laboratory facilities. These can be virtual or remote. The virtual ones simulate the functionality of a real laboratory and send the simulated experimental data to the user. The remote ones have real equipment as backend, on which the measurements take place.

The remote laboratories present a high interest for the ones that learn, because they have the possibility to experiment on real equipment from home or from work, using equipment located into a university laboratory for example. This facility is important in understanding the processes from the practical point of view. The equipment, often expensive from the laboratories will become accessible to the wide public.

As principle of functionality, electronic driven systems can accept remote commands sent by third party applications, through Internet, as this is an electronic environment. An important issue for the user is what can he obtain from a laboratory, using such a communication system. Obviously he will not be able to access the labo-

ratory as if it was there. Here, the user interface comes in, which is usually represented by a web page. The way in which the page is made and what it will allow the user to do, has a great impact on the learning and understanding of the processes that take place in the real laboratory.

For developing the the remote and virtual laboratories user interfaces we use Java.

The software development is nowadays one of the most extended domains, and sure, the most dynamic. In such a dynamic environment as IT, time is essential, and this can be noticed in the general IT market trends over the world. Everybody wants better productivity when it comes to developing software, which is basically translated into obtaining similar products over a significantly shorter period of time.

The answer to better software development productivity is Java because the many features it encapsulates frees the programmer from a lot of work, when compared, for instance, with C++.

In client applications the execution speed is not the most important factor, but instead, the portability is a very important issue. As Java is built to be portable, this meaning that the same Java application, written and compiled on a specific operating system, will be able to run on other operating systems without any changes - a great advantage.

So, we have decided to use Java as development tool for the following main reasons: it is suitable for web-based applications, it is portable and it allows much better productivity, along with developing robust applications.

2 The Java language

Sun Microsystems describes Java as a "simple, object-oriented, interpreted, robust, safe, architecture independent, portable, high performance, multithreaded and dynamic" programming language.

Each feature mentioned above is an important part of the Java programming language, as well as a critical request for web programming. The combination between these features makes Java a powerful and useful programming language that supplies the programmer with the tools he needs to easily create powerful programs for the distributed applications environments of today.

On the Internet, the Java programs are called *applets*. The applets are Java applications that are embedded into HTML web pages and can be loaded into a Java capable web browser, at a click of the mouse.

Before the browser's Java runtime interpreter loads and executes the applet code, it checks the code for integrity. Java is more than a simple tool for writing applets. It is a new and powerful programming language.

Now, the software companies clients ask for shorter development time, competitive products, scalable and portable. The speed problems are no longer a prime concern for software programmers or users. The hardware evolution eliminated the concern for system resources, pointing the consideration towards quality and capability.

Because of its extended portability and easy programming, Java is the optimal choice when developing client applications. Although Java programs are slightly slower than C++ programs and have some disadvantages concerning the use of plat-

form specific features, Java is the preferred programming language for many software developers around the world.

Due to the fact that there exist a lot of operating systems (Windows, Unix, Linux, MacOS, etc) the client applications, if we would use C++, should be written for every platform. Using Java, they must be written only once, on any of the platforms and will be runnable on any other platform without changes. This advantage, along with others offered by Java, clearly places it on the top of client applications programming language.

Java, be it as applications or applets, is certainly the most recommended programming language for designing and developing client applications, where the execution speed is not critical and no platform dependent resources are needed.

3 Considerations on the applets developed in the virtual ElectroLab project

3.1 The web-page

All the Java Applets are integrated as objects into web pages (a separate web page for each applet). Because the applet is used only as tool for posting measurement requests and for viewing the measurement results, it does not contain information about how does the experiment take place or about the meaning of the values sent or received. In order to make the applet easy to use by uninitiated students, we provide basic information on how to use the applet and which is the meaning of the values in the experiment in the same web page (which contains the applet).

There are two possibilities to present the descriptive text:

A) On the top of the page, before the applet. In this case, the applet is at the bottom of the page and the user must scroll over the descriptive text in order to reach the applet and make the experiment.

B) After the applet. In this case, the applet is at the top of the page, and the descriptive text after the applet. In this case, the user will read the information only if necessary.

This descriptive text has two meanings:

- as laboratory guide for first experiments;
- as help afterwards.

The text placed before the applet is useful in the first stage, because the student can see that there is information about the experiment, but it can be uncomfortable for students that already know how to use the applet, because they have to scroll the text everytime they enter the page. The text presented after the applet is better fit for students that know how to use the applet. It can be a problem for students that don't know how to use the applet. They may not see that there is a descriptive text after the applet.

The e-Learnig course that corresponds to the laboratory experiment can be accessed very easy from the web page that contains the applet, because there is a link to the course topics page.

Until now, the applet was sending a request to the server and after that it was entering a loop into which it waited for a response to arrive. After the response was received, the applet showed the graphic representing the measurement. Because the physical measurement process can take up to a minute for a single request and because there can be, for example, 30 requests waiting to be solved, the user would have to stay connected to the server 30 minutes waiting for the response. This was considered as being a problem and so, we have splitted the process of posting a request and seeing the results into three, separate and asynchronous operations:

- posting a request;
- viewing a list of posted requests;
- viewing the graphics for a certain item of the list.

3.2 Implementations made using the asynchronous mechanism

The web page team has developed a database using SQL Server used as base for the final version of the system. Using this database, we have changed the functionality for the following three applets:

- Electromagnetic Materials;
- Electronic Circuits;
- Electrical Drives.

Electromagnetic Materials Laboratory

The web page that contains this applet presents the descriptive text at the top of the page and the applet at the bottom, which practically represents the first approach presented at the beginning of chapter 3.1.

The first view (fig. 1) presented by the applet is used for posting measurement requests. When pressing "Send" a request is posted on the server.

When pressing history, the applet shows a table containing all the posted requests, presented in figure 2. When the user presses "View" the applet will show the result, presented in figure 3, for the selected request.

If the "Refresh" button is pressed, the table which contains the list of requests is refreshed so that it shows the current requests configuration.

In figure 3 are presented the measured values for the selected request (in the History view). Three types of graphics are available (B(H) presented in the screenshot below, B(t) and H(t)).

The table shows the individual measured values and the textbox in the lower-left corner shows the measured graph points.

The applet checks the values entered by the user and does not send the request if the input values are not in the valid range. The user is prompted if he enters invalid values.

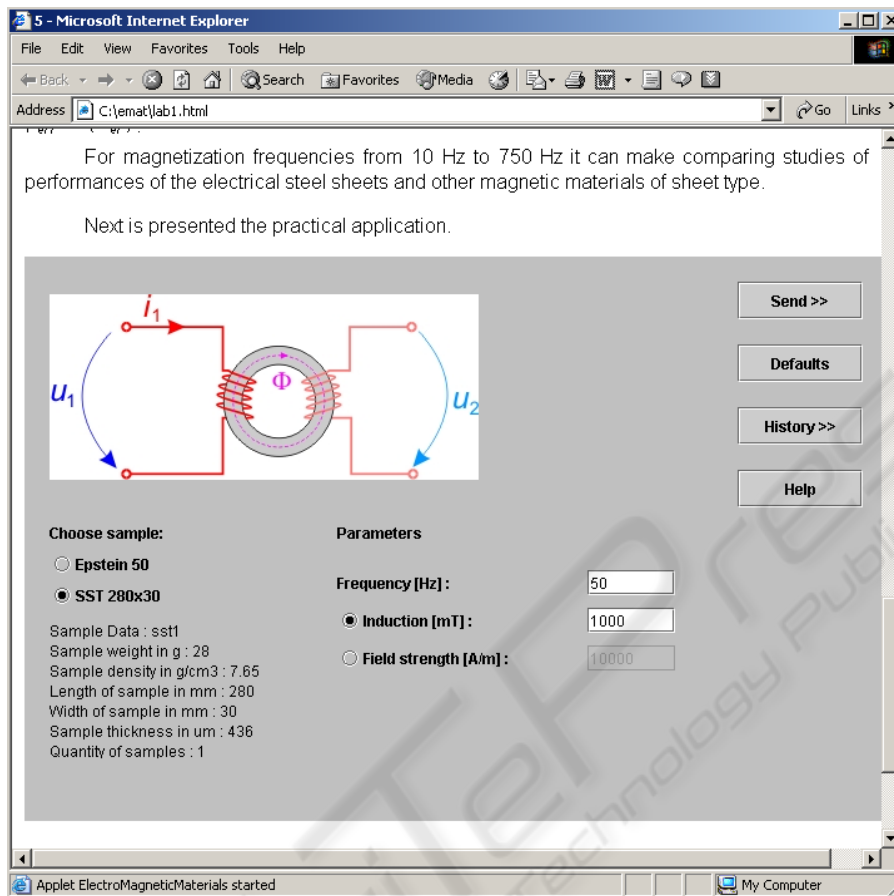


Fig. 1. The main view of the Electromagnetic Materials.

Fig. 1. presents the layout of the main applet view. The picture located at the upper-left corner shows a simplified schema of the measurement equipment. The two radio buttons below the schema can be used to select the type of sample. Below the radio buttons are printed the specifications for the selected sample.

The user can choose to enter either the value for induction, either for the field strength, by choosing the appropriate radio-button located in the lower center of the view.

The "Help" button will show a HTML page containing the description of the laboratory and documentation on how to use the applet to take experiments.

5 - Microsoft Internet Explorer

Address: C:\yemat\lab1.html

Review a measurement:

Sample	Frequency[Hz]	Induction[mT]	Field strength[A/m]	Date and Time
SST 280x30	50	1000	-	2004 April 15 01:56:27
SST 280x30	50	1000	-	2004 April 15 01:56:40
SST 280x30	50	1000	-	2004 April 15 01:59:53
SST 280x30	50	1000	-	2004 April 15 02:38:24
SST 280x30	50	-	555	2004 April 17 12:02:54
Epstein 50	50	565	-	2004 April 17 12:05:08
SST 280x30	50	1000	-	2004 April 17 01:20:37
SST 280x30	50	1000	-	2004 April 17 01:20:52
SST 280x30	50	500	-	2004 April 17 02:24:52
SST 280x30	50	1000	-	2004 April 17 04:56:09
SST 280x30	50	1000	-	2004 April 17 04:56:24
SST 280x30	50	1000	-	2004 April 17 04:59:07
SST 280x30	50	-	10000	2004 April 17 05:01:11
SST 280x30	50	-	10000	2004 April 17 05:11:19

After posting a request press Refresh to get response!

Fig. 2. The history view for the electromagnetic materials.

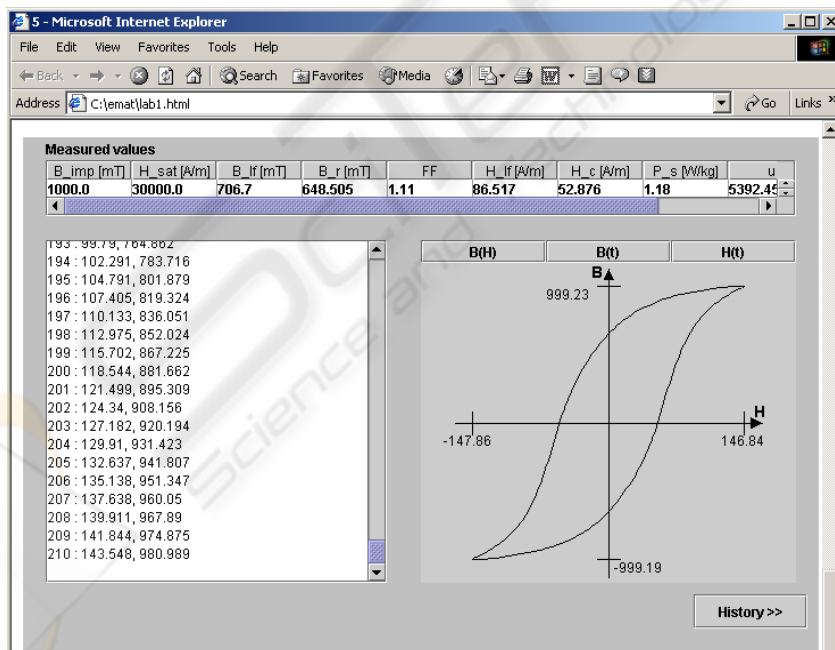


Fig. 3. The result view for the electromagnetic materials showing the graph and measured values.

By using the view in fig. 2., a user can see all the requests he made over time. Just as in mail systems, the requests and corresponding responses which were already viewed are written normally. The new records are marked with bold. The user can select one of the records and the selected record will be shown with a cyan background. When a record is selected, the view button will become active, otherwise being inactive. The Refresh button will bring the new records from the database.

For knowing which measure was viewed and which not, when pressing the "View" button, the information about the response for the selected request is being retrieved from the database. The application also updates the request status (as being viewed) in the database when making the request.

Electronic Circuits Laboratory

The web page that contains this applet presents the descriptive text at the end of the page and the applet at the top, which practically represents the second (B) approach presented at the beginning of this material.

The main view (fig. 4) of the applet is used to post requests. The user can choose which source (U_b or U_c) will vary its voltage and the values for the voltages (min , max and $step$). After pressing "Send request" the applet will send the request to the server and will present the history view, which is also accessible by pressing the "History" button. The type of transistor can be only "bipolar npn" because the physical circuit does not support the other types yet.

The history view shows a list with the posted requests. The user can choose a request and click "View the selected request" to view the graphic results for that request. Clicking "Refresh the table" will refresh the information (show the new status from the database).

When the user clicks "View the selected request" the applet gets the result corresponding to the selected request from the database and opens a new window into which a graphic representation is drawn.

Electrical Drives Laboratory

The main view presented in figure 5 is used to post the requests. The input values are for "number of samples", "Km" and "KI". Pressing "Send request" will send a request and open the history view, which is also accessible by pressing "History".

The Electrical Drives web page is organized like the Electronic Circuits web page. So, the descriptive text is placed at the bottom of the page.

The history shows, similar to the Electronic Circuits applet, a list with the posted requests (figure 6). The list is obtained by making a query into the database.

After selecting a request, the user can see the interpreted results as graphics by pressing the "View the selected request" button in figure 6. The applet will open a window into which the user can choose to see one of the two available graphs (speed variation or voltage variation).

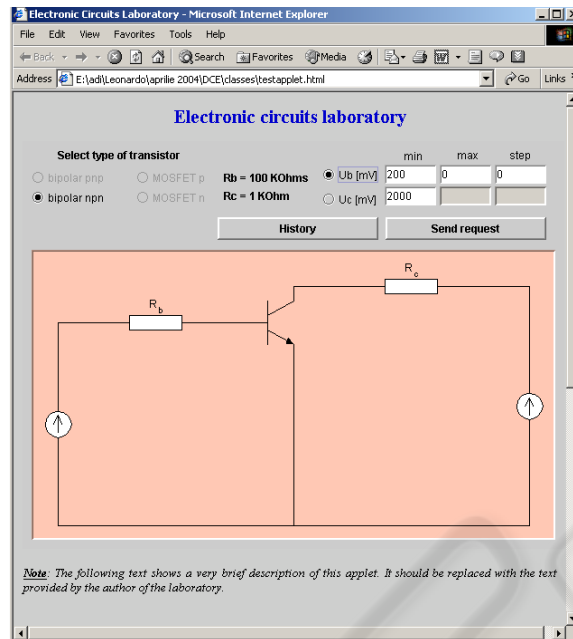


Fig. 4. The main view of the Electronic Circuits Laboratory.

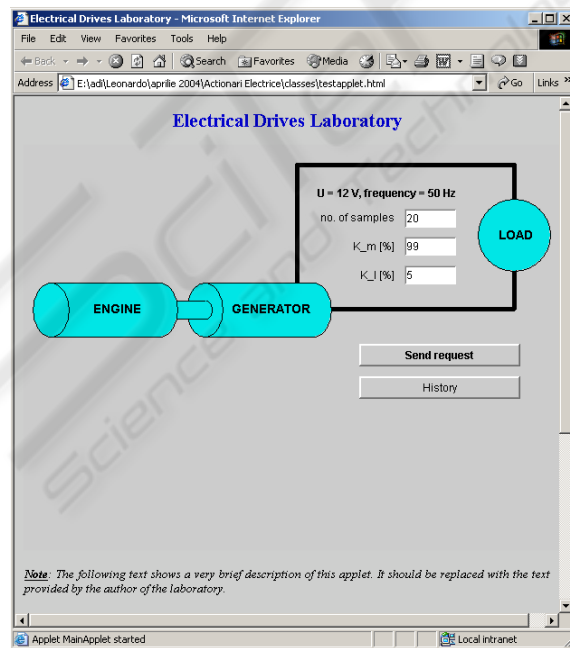


Fig. 5. The main view of the Electrical Drives Laboratory.

Electrical Drives Laboratory

Choose one of the following request posted on the server and click "View" to see the graphs

samples	Kl	Km	Date and Time
20	99	5	2004-04-17 17:55:20.062
20	99	5	2004-04-17 17:58:57.015
20	99	5	2004-04-17 18:04:42.843
20	65	5	2004-04-17 18:10:17.734
55	6	5	2004-04-17 18:12:05.453
20	99	5	2004-04-18 12:05:10.614
20	99	5	2004-04-20 14:13:42.801
20	99	5	2004-04-20 15:09:21.531

Fig. 6. The history view of the Electrical Drives Laboratory.

4 Conclusions

The use of Java applets as web-based user interface for real laboratories increases the usability of remote instruments.

The student's capability to understand the phenomena that take place in the real experiments is improved by the use of real data.

Developing intuitive user-friendly interfaces we achieve the goal of making the training material interesting for all the students and transforming the act of learning into a pleasure.

Using the Java technology we provide a good portability on various platforms for the applets, so the users are not restrained to use a specific operating system.

Java offers a fast and performant way of developing web-based applications with a wide area of accessibility.

References

1. Taylor, A., 1997. *JDBC Developer's Resource*, Informix Press.
2. Horstmann, C., Cornell, G., 2001, *Core Java 2 vol. I and II*, Sun Microsystems Press.

3. Weiner, S., Asbury, S., 1998, *Programming with JFC*, Wiley Computer Publishing.
4. Bassem, A., Khalid H., Samer, A., Abul-Humos, A., 2001, *Remote labs: Electrical element characterization experiment*, International Conference on Engineering Education
5. Cassini, M., Prattichizzo, D., Vicino, A., 2002, *E-Learning by remote laboratories: A new tool for control education*, Siena University, Italy
6. Khamis, A., Rivero, D.M., Rodriguez, F., Salichs, M., *Pattern based Architecture for Building Mobile Robotics Remote Laboratories*, Carlos III University of Madrid
7. Trevelyan, J., 2003, *Remote Laboratories and Team Skills in Mecatronics*, Mecatronics and Machine Vision in Practice 2003 Conference, Western Australia.

