

## ICT USE FOR REMOTE EXPERIMENTS IN ELECTRIC AND ELECTRONIC DOMAIN

**Sorin-Aurel Moraru**

**Adrian Pelcz, Ionut Diaconu**

**Dominic Mircea Kristaly, Catalin Bujdei**

Automatics Department, "Transilvania" University of Brasov, M.Viteazu Street, no.5, 500174, Brasov, Romania, phone/fax: +40 0268 418836,  
smoraru@vision-systems.ro  
apelcz@vision-systems.ro, idiaconu@vision-sytems.ro  
kdominic@vision-systems.ro, cbujdei@vision-systems.ro

*This paper refers to the e-learning domain both from the virtual and remote laboratories point of view and from the security one. 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 the 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 some implementations in the pilot project "VirtualElectroLab" of the "Leonardo da Vinci" program that have user interfaces based on the Java Applet technology, for accessing the laboratory resources via Internet. When it comes about accessing an application remotely, that is from another computer than that where the application resides, the security should always be of concern. Because the connection between the two computers is done through a physical space, on which we may not have any control, whether that connection is through wires or wireless, we must be on guard of any possible identity and information stealing.*

**Keywords:** ICT, remote, e-Learning, Java, databases, information security

### 1. INTRODUCTION

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 laboratory 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 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 marked 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++ [1].

## **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 [3].

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.

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 platform 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.

## **3. APPLETS DEVELOPED WITHIN THE *VIRTUALELECTROLAB* 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).

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. We have split 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 implemented the functionality for all the applets, corresponding to the following electric and electronic domains: Electrical Drives, Automated measurements, System Theory and Automatic Control, Domatics, Databases with Applications in the Electrical Domain, Electronic Circuits, Electric Circuits, Electrotehnic Materials, Electrical Machines. We will exeplyfy with the laboratory of 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 (figure 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 the figure 2. When the user presses "View" the applet will show the result, presented in the figure 3, for the selected request.

If the "Refresh" button is pressed, the table that 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 put 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.

The figure 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.

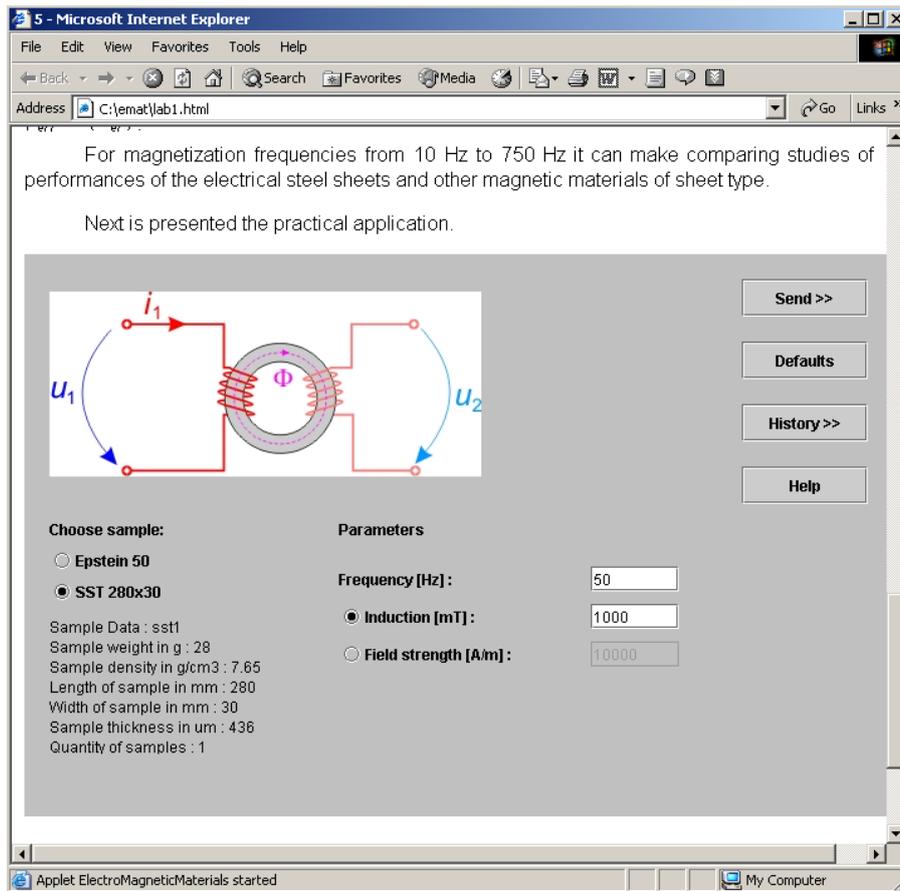


Fig. 1. The main view of the Electromagnetic Materials

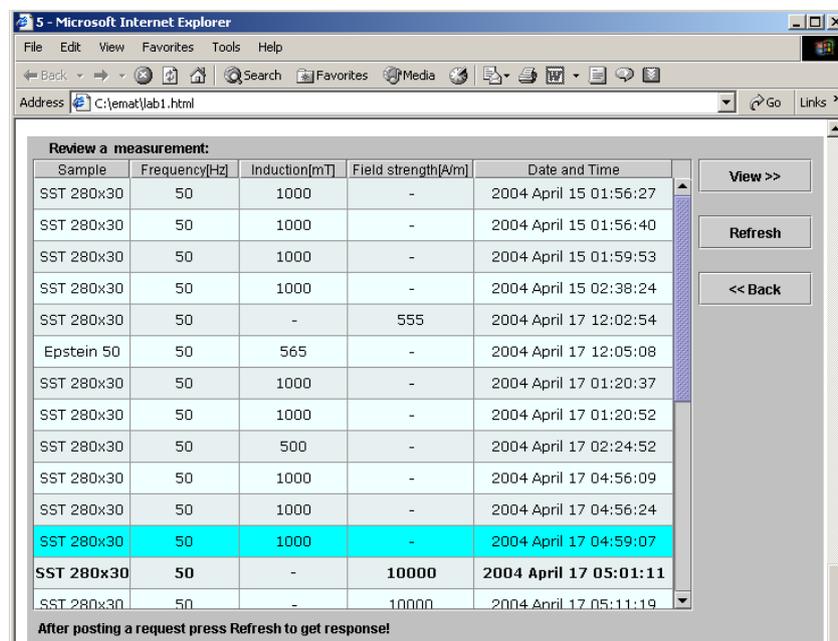


Fig. 2. The history view for the electromagnetic materials

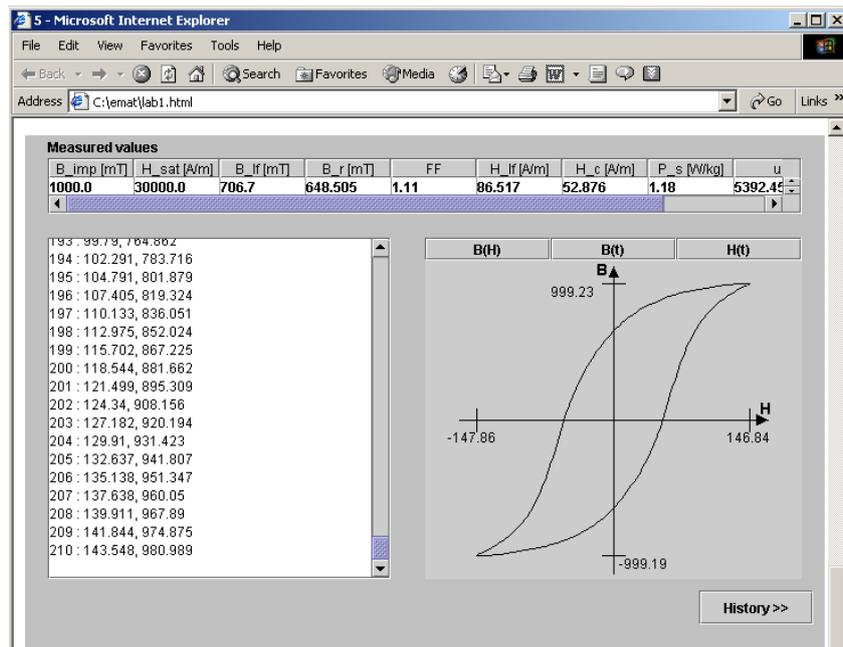


Fig. 3. The result view for the electromagnetic materials showing the graph and measured values

By using the view in figure 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.

#### 4. AUTHENTICATION AS A MAIN PART OF THE SECURITY ASPECTS

Any communication is realized between two or more subjects. Each of them must identify itself to the others, in brief or in details, before being able to participate to that communication [6]. After that, a continuously switch between emitter and receptor roles takes place among the subjects.

In our case we have a communication between the client computer, which makes requests and two servers that respond: the *Learning Management System* (LMS) server and the *VirtualElectroLab* (VEL) server.

##### 4.1. The client authentication

In order to maintain a communication with more than one client, the application uses a session mechanism, meaning that it allocates resources (variables, threads, memory etc.) for each one of them, and an identification, which distinguishes one client from the other. Every time a client makes a request, it has to reveal its own session ID, letting the server know its identity and the stage of the communication.

For initiating a communication, the client must send his username and password to the LMS server. The server will verify the existence of this user into the database and depending of the result, will allow the client to connect to the features of the application or not. In case the login succeeds, the server allocates new resources for this new client and sends the ID of this newly created session back to the client. As mentioned, the client will use this session ID every single time it makes a request to the server, identifying itself among all others connected users.

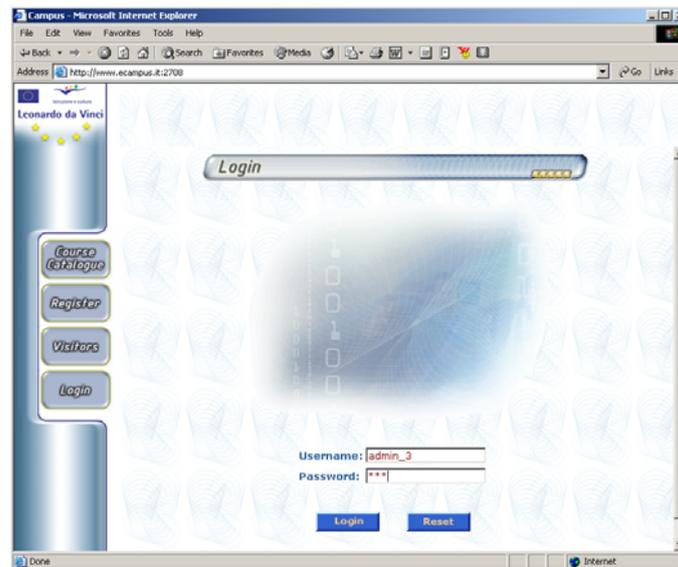


Fig. 4. Login page for the LMS application

The use of the pair username / password has another reason too. It allows the access of the client to different features of the application, depending of his role: administrator, teacher, student etc. (fig. 4).

The VEL server has not a login mechanism of its own, but it will use the login feature of the LMS server. Thus, if a client accesses the VEL server without passing through the checking made by LMS through username/password pair, it will be rejected. The VEL server knows if a client has successfully passed the login, by looking at the session ID provided by the client along with the request.

#### 4.2 The server authentication

Not only must the client identify itself, but also the server. Because the application of VEL is very complex, a part of it is executed on the client machine. For this reason, a Java Applet has been chosen as solution. Within it several computing operations are performed on the client part. In this way, we can reduce the network traffic, because instead of sending multiple requests to the server for different manipulation of data, we can receive the data in the Java Applet client and then all the wanted actions regarding that set of data can be done locally, without any new request to the server.

But this facility offered by the applet raises another issue. The Java Applet, being capable to execute commands on the client machine can executed unwanted commands in order to harm the client's computer.

Because of this potential problem, an applet is not allowed to perform freely, but instead it is subject to a so-called “sandbox”, that is a delimited set of permitted actions, anything else outside this sandbox not being available for execution. Among the actions prohibited, are accessing client's files, opening channels of communications to other computer than that where the Applet originated etc.

However, if an application really needs to access some resources on the client machine, such as the file system, the printer etc., there is a mechanism through which this can be accomplished. This mechanism is called “Applet signing”. By signing it, an applet requires, at its initialisation, the acceptance from the user to execute actions outside the sandbox [7].

Therefore, when one loads into a browser a Java Applet, he must know who the server is in order to make a clear judgment about if he trusts it or not to let it run commands on his computer. That brings us to the server authentication matter. For this very reason a server must be able to identify itself in order to make the client trust that it is who it claims it is and allowing the clients deciding if they trust it or not. In this way, a third party server cannot claim to be someone else for deceiving the users and running malicious code on their machines.

#### **4.2.1 Certificates**

The first thing in authentication is the creation of a certificate that describes the entity of the claimer. It is made of the main details of the entity: name of person, company name, department name, location, country. This would be the equivalent of an ID of a person in the real world.

Now, some authority in the domain must authenticate this certificate, in the same way that in the real world an authority issues a driving license or an ID. In case of digital signing, there are some companies like DeutschePost, Thawte, Verisign or Ecquifax each one known as Certificate Authority. These companies are entitled to validate those certificates by priory verifying the credentials of the claimer. Then, it applies its own signature over the entity's certificate, thus validating it.

#### **4.2.2 Packing the Java Applet**

The next thing after having the signed certificate, is packaging all the components of the Applet in order to signing it. This is done with the JAR tool.

#### **4.2.3 Signing the Applet**

The third step represents the actual signing of the Applet, now packed into a Java ARchive. The tool used for the operation is *signtool*, provided with every Java Development Kit.

When a user loads a web page that contains a signed applet, a warning is presented to the user (fig. 5).

The user can see the details about the certificate by clicking “More Details”.

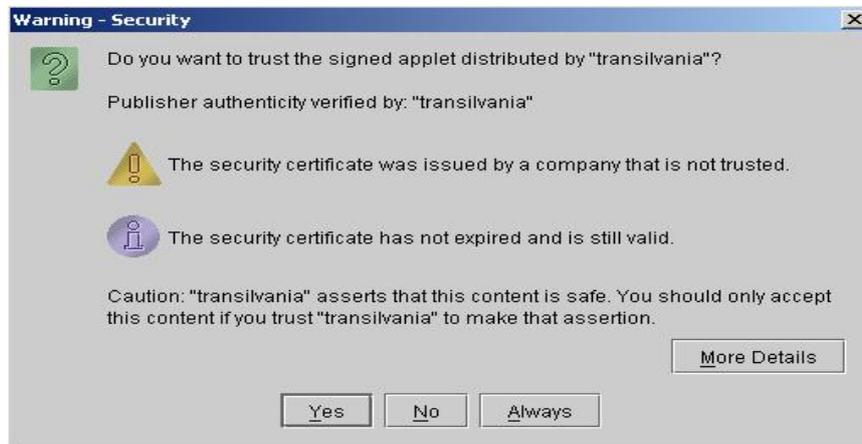


Fig. 5. Warning of loading a signed applet, which requires additional privileges for the execution of commands outside the sandbox

## 5. 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 performing way of developing web-based applications with a wide area of accessibility.

## 6. REFERENCES

- [1] Moraru S.A., I. Diaconu, A. Pelcz, S. Leahu, *The Database and Architecture of Virtual and Remote Laboratories*, 1st International Conference on Informatics and Control, Automation and Robotics, p. 38-42, ISBN 972-8865-14-7, august 2004, Setubal - Portugal.
- [2] Taylor A., *JDBC Developer's Resource*, Informix Press, 1997.
- [3] Horstmann C., G. Cornell, *Core Java 2 vol. I and II*, Sun Microsystems Press, 2001.
- [4] Cassini M., D. Prattichizzo, A. Vicino, *E-Learning by remote laboratories: A new tool for control education*, Siena University, Italy, 2002.
- [5] Trevelyan J., *Remote Laboratories and Team Skills in Mechatronics*, Mechatronics and Machine Vision in Practice 2003 Conference, Western Australia, 2003.
- [6] [http://ecommerce.mit.edu/class/private/crypto\\_summary.html](http://ecommerce.mit.edu/class/private/crypto_summary.html)
- [7] <http://java.sun.com/developer/onlineTraining/Security/Fundamentals/Security.html>
- [8] <http://www.openssl.org>