12th IMEKO TC4 International Symposium
Electrical Measurements and Instrumentation
September 25–27, 2002, Zagreb, Croatia

REMOTELY ACCESSIBLE LABORATORY FOR ELECTRONIC MEASUREMENT TEACHING

G. Canfora, P. Daponte, S. Rapuano

Dipartimento di Ingegneria, Università del Sannio, Benevento, Italy.

Abstract – The paper presents a remotely accessible laboratory realised for didactic aims at the University of Sannio, Italy. The laboratory is based on a software framework, with modularity characteristics that allows the insertion of new applications or the modification of the realised ones. Thanking to its implementation in Java and C++ with a CORBA communication layer, some instruments have been made available to the students of electronic measurements courses through a simple web browser in the form of Virtual Instruments.

Keywords: remote laboratory, measurement teaching, Java, VI.

1. INTRODUCTION

The wide spread of Internet based facilities had a big impact in the measurement related tasks. In particular, the World Wide Web makes available the measurement resources and data worldwide by requiring a minimum connection cost. Moreover, Internet popularity produced the development of a great number of tools enabling the realisation of flexible customized solutions for the specific remote measurement tasks.

Some papers can be found in literature using Internet as a communication channel enabling the realization of remote and distributed measurement systems (DMS) [1-6]. Also from the didactic point of view much work has been done. The potentiality of remote teaching [7] and, in particular, the use of Internet as a channel to reach the students or workers to their homes has been soon recognized [8-14]. At now, four main approaches have been following for the remote teaching, (i) Web based lectures and seminars, sometimes interactive, provided by hardware or software producers, mainly directed to professionals that want to reduce the start-up time for a new application [15], (ii) Web support to University courses, including slides of the lessons and exercises [16,17], (iii) simulation of actual experiments to be executed either remotely or on the user's PC [18-21], and, more rarely, (iv) remotely accessible laboratories, where the students can access the actual instrumentation through a Web page [1-3,22]. Although the main efforts of the researchers are devoted to the DMS oriented to the research or industrial applications, the necessity of a common education system based on the Web is a main topic for the European Universities; and a thematic network has been realized with this aim [23].

This paper focuses on the problem of realizing a really effective remote laboratory oriented to the teaching of measurement science. This field more than others, in fact, requires a huge practical experiencing in order to assure a good knowledge transfer from the teacher to the students. The students should achieve such experience by working in actual conditions and on the actual instruments. Usually, indeed, the sophisticated and expensive measurement instrumentation involved makes it difficult to keep the technical staff up-to-date. Moreover, there is the necessity for repeating the same experience many times in order to make all students able to operate the measuring instrumentation [24-27], which makes the technician shortage even more severe.

To design the remote laboratory, the parallel with the DMS design can be exploited by taking all the possible advantages from the technologies used for the development of industry and research oriented DMSs.

Among the problems that should be faced in the design phase of a DMS, the following three have a great importance (i) the system architecture, (ii) the number and the type of instruments to be used, and (iii) the software solution to be implemented.

The first two questions are almost trivial, in the case of a remote laboratory with didactic aims. The architecture, in fact, is almost pre-defined by the problem itself: to distribute to the larger number of students the resources of one or a few of laboratories. A simple client-server structure, with one or a few servers and many clients, is the simplest answer. From a deeper observation, a less trivial problem arises: how to assure that the resources will be shared in an effective way, without overloading the servers and serving all the queries. This problem concerns the software. The present paper addresses the problem by using a centralized software management system.

The number and the kind of instruments to be used depend on the experiment that the teacher chooses to implement within his/her course, so it has not to be pre-defined or fixed a priori.

The software solutions to be chosen constitute the main difficulty of the task, as the software should address the following questions:

- to manage the students' access to the laboratory;
- to set the connections among the servers and the measurement instruments;

– to set the communication among the clients and the servers;
– to assure the data integrity during the experiments;
– to provide one or more graphical user interfaces for the instruments within Web pages;

with the following requirements:

– low set-up cost and development time, from the provider side;
– low connection cost from the client side, that is, low navigation time, considering a typical dial-up connection speed of about 56 kb/s;
– high flexibility and scalability, that is, the possibility of adding or subtracting experiments without rewriting all the code;
– high interchangeability, that is the use of data formats that are not proprietary ones.

More users could simultaneously require executing the same or a different measurement experiment on the same laboratory instruments. Consequently, a need for the correct handling of measurement request concurrence arises in order to avoid system faults due to access conflicts.

The above quoted problems have been faced by developing a remote laboratory facility at the Laboratory of Signal Processing and Measurement Information (LESIM) of University of Sannio. The target has been achieved by following the approach developed for the DMSs [28-31] and the remote teaching [3,32]. In particular, by adding the Distributed Object Technology and the wide use of Java, a software framework and some applications have been realised. In comparison with the previous works in this field, they provide a better system independence and a greater flexibility. The developed laboratory is oriented to educational purposes and provides both hypertexts and a set of related basic experiments on topics included in electronic measurement courses. In the experiments, the user can perform remote measurements and download the results for local processing by means of a Web browser without any proprietary client application. In the following, first the hardware and software design solutions are presented, then some examples are given with the aim of inviting the reader to access the Internet site (http://lesim1.ing.unisannio.it) to get a deeper knowledge of the features of such laboratory.

2. SOFTWARE SOLUTIONS

In order to allow an easy user interaction with the physical instrumentation installed in the LESIM, the application software has been implemented in the form of Virtual Instruments (VIs). In particular, a great interchangeability and an easy access through Internet have been obtained by implementing the instruments' Graphical User Interfaces (GUIs) in Java. This approach aims to produce very easy and user-friendly virtual front panels, that actually are GUIs for already existing instruments.

In order to implement DMSs Java has been often chosen as a platform independent, object oriented, non-proprietary alternative to the classical development environments, such as LabVIEW [4-6]. For example, the solution proposed in [30] uses an architecture based on a measurement server, managing the instrumentation and collecting the results, and a web server for exporting them over Internet. The communication between the client and the server can be implemented by using distributed object technology, such as Java RMI (Remote Method Invocation) or CORBA (Common Object Request Broker Architecture) standard [31,32]. A DMS development environment based on Java has been proposed in [33]. It provides a library of instrumentation components written in Java, thus enabling the realisation of VIs in a graphical programming way. It uses a Microsoft Visual Java feature to implement the communication with the instrumentation drivers, written in C++. This development environment, in fact, makes the driver functions accessible through a call from a pre-compiled .dll library. However, this choice requires the execution in a Microsoft operating system, thus removing the platform independence that is the most attractive characteristic of Java. Moreover, at now, the components provided are not enough for competing in the fields of scalability, design freedom and development time, with proprietary solutions implemented in LabVIEW or LabWindows/CVI.

2.1 Distributed Object Technology

Distributed software technology includes three technologies that have been synergically merged, namely object oriented programming, distributed systems, and the web and its accompanying tools. The combined use of these technologies is rapidly changing the way in which complex systems are conceived and developed, generating a new paradigm of computing at the hearth of which are interoperable objects. To provide the necessary background and perspective on the present work, in this section the object technology from two points of view, namely the middleware and the programming paradigm and language will be briefly reviewed. In particular, CORBA for middleware and Java applet technology for Web programming will be discussed.

Middleware: Common Object Request Broker Architecture (CORBA)

As object oriented paradigm became more and more popular through the 1980s, there was a growing interest in combining the concept of objects with the concept of transparent distributed computing. Objects, with their inherent combination of data and behaviour and their strict separation of interface from implementation, offer an ideal package for distributing data and processes to end-user applications. In the early 1990s, the OMG (Object Management Group) defined a standard, CORBA, for the distribution of objects. Fundamentally, CORBA is a distributed client/server platform with an object-oriented spin. The idea is to provide an object-oriented programming model for distributed computing to programmers that is as close as possible to programming with ordinary local objects. CORBA addresses issues related to interface, registration, databases, communication, and error handling. When combined with other object services defined by OMG's Object Management Architecture (OMA), CORBA becomes a middleware standard that facilitates full exploitation of object technology in a distributed system. CORBA hides most of the difficulties of distributed programming by offering the following key properties:

- Location/server transparency: Clients do not know whether the target object is local to its own address space, if it is implemented in a different process on the same machine, or it is implemented in a process on a different machine. In addition, clients do not need to know on which server the objects are implemented.
- Language independence: CORBA is concerned with interfaces of objects and does not specify implementation. For example, a Java client can call a C++ implementation without being aware of it.
- Architecture/operating system independence: clients are unaware of the architecture and the operating system that are used by the server. Indeed, CORBA ensures also independence of the communication protocol and transport layer.

Web programming: Java and Applets

A key concept associated with the Web programming is that of "mobile objects" or "executable content," which is possible with Java, and particularly Java Applets. Applets are executable on all platforms, including UNIX boxes, Windows/Intel machines, and Macintoshes, by using a Web browser. The Web, thus, becomes the delivery vehicle for programs that can be executed on a client machine and that can reach back to the server for specific operations. This power does not derive from the language per se, but from the architecture-neutral approach used by Java. Rather than producing machine-specific instructions, Java is translated into bytecode. The Java virtual machine (JVM), which lives in an operating system or a Web browser, translates the bytecode into the machine-specific instructions. This gives Java its platform independence, making Java applications completely portable. While the execution of applets is somewhat slow, there are "just-in-time" (JIT) Java compilers that compile the bytecode on the client machine before executing them, resulting in improved performance. Another key component of the Java technology is the application programming interfaces (APIs) that are standard, but separate from the language. These include the Abstract Window Toolkit (AWT) that provides a complete set of classes for writing graphical user interfaces. Finally, Java supports distributed object programming by means of RMI. However, whilst the basic ideas of RMI and CORBA are similar, CORBA allows the co-operation between pieces of code written in different programming languages, while RMI is limited to Java to Java communication. This difference is very important where it is necessary to merge C++ written measurement procedures with a Java written GUI in order to realize a single VI.
CORBA and Java each offer substantial support for the construction of distributed systems; together, they present application developers with unprecedented capability to build new ones.

3. THE REALISED REMOTE LABORATORY

The system architecture is based on a management software that provides a friendly environment to the user and a list of links to the VI interfaces for the specific experiments. Moreover, it provides a suitable framework to the development of new remotely accessible experiments, thanking to a highly modular architecture.
At now, the remote users may exchange information by displaying hypertext documents (Web pages) written in HyperText Mark-up Language (HTML), by means of the Internet, WWW and common browsers. The client computer (i) accesses files located on the LESIM web server; and (ii) uses browsers to decode the HTML in those files for displaying documents as texts, data or graphics and for defining hypertext links among documents [1-3].
The use of light Java *applets* allows the user to download the VI GUI with no need for proprietary run-time engines. In this way, several users can simultaneously browse an experiment page by means of a commercial Web browser, usually included with no additional cost in actual operating systems such as Internet Explorer™ or Netscape™.
Both the managing and application software have been designed with the aim of maximising accessibility, portability and cost saving. They have been developed on the Windows NT environment [34] that allowed the web server to be easily and quickly implemented.
Conventional desktop Pentium-based personal computers have been used as low-cost hardware platforms. A PC has been used to implement the system server, while the others have been used as measurement servers with the tasks of controlling and collecting data from the actual instrumentation. The system server and the measurement ones are linked through an Ethernet LAN.
Physical instruments have been linked to the measurement servers by means of a MXI-II interface, an RS-232 interface and a Data Acquisition Board.
The following experiments have been realised and currently accessible from the LESIM site:

1) Remote control of VXI instrumentation. A VXI 4250 waveform tester and a VXI 73A-243 arbitrary waveform generator are available to the students in order to take experience with waveform generation and acquisition problems.
2) Remote programming of FIR filters, by using a TMS320C50 DSP from Texas Instruments.
3) Remote control of an industrial process emulator, accessible via a PCI 6023 data acquisition board from National Instruments.

Each experiment is provided with some tutorial pages, and some theoretical hints, in order to recall the information required to carry out a correct experience.
In the next paragraphs, the managing software features and the experiments are briefly described.

3.1 The laboratory management software

The main task of the management program is to receive the user requirements and either to provide hypertexts with tutorial aim or to activate the instruments of the automatic measurement station linked to the server.

As mentioned above, some problems could arise when the users try to interact with the instruments. Several users, in fact, can simultaneously input measuring requirements by means of HTML forms. A possible conflict due to simultaneous access to the same instrument has been avoided by arranging the server to manage the concurrence. A queue is established among the logged user requests and each of them is served according to the first-in first-out concept [1-3].

The managing software has been developed as a web application by using the ASP technology (Fig.1). It is based on a database structure that includes data from the physical instruments connected to the system. From the main page the users are enabled in accessing the system for educational or administration purposes, as well as the implemented experiments.

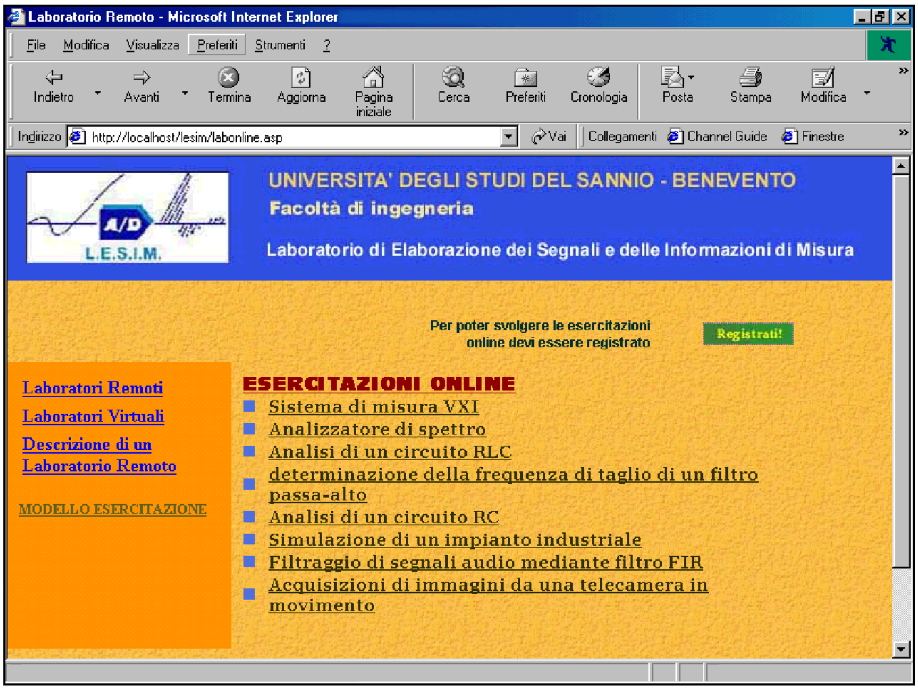

Fig. 1. Main access page to the remote educational activities of the LESIM laboratory.

In particular, the developed software:

- manages the user queries verifying if the required instrument is already used;
- manages the user queues;
- verifies the status of the physical instruments connected to the system, enabling the use of the working ones only;
- enables the insertion of new instruments into the system by means of a web-based interface, with restricted access for the system administrator only;
- permits to move a VI from a PC to another, connected to the WAN, enabling the realisation of a geographically distributed architecture;
- allows to the administrator an easy enabling or disabling of single experiments.

3.2 Remote control of VXI instrumentation

A first experience provided to the students deals with waveform generation and acquisition problems.

Two VXI instruments have been used for such experiment, a VXI 73A-243 12 bit waveform generator and a VXI 4250 8 bit waveform tester. The physical instruments are managed by means of C++ programs while the corresponding VI has a Java written GUI. The two parts communicate through a CORBA layer. As it can be seen in Fig.2, from the generator side, the users can set the type of waveform to generate, its amplitude and frequency. On the acquisition side, an oscilloscope-like panel has been provided, allowing in addition the choice of the sampling frequency. This solution allows to the users to take experience with the problems concerning practical acquisitions, such as aliasing or waveform truncation when the input range is lower than the input voltage.

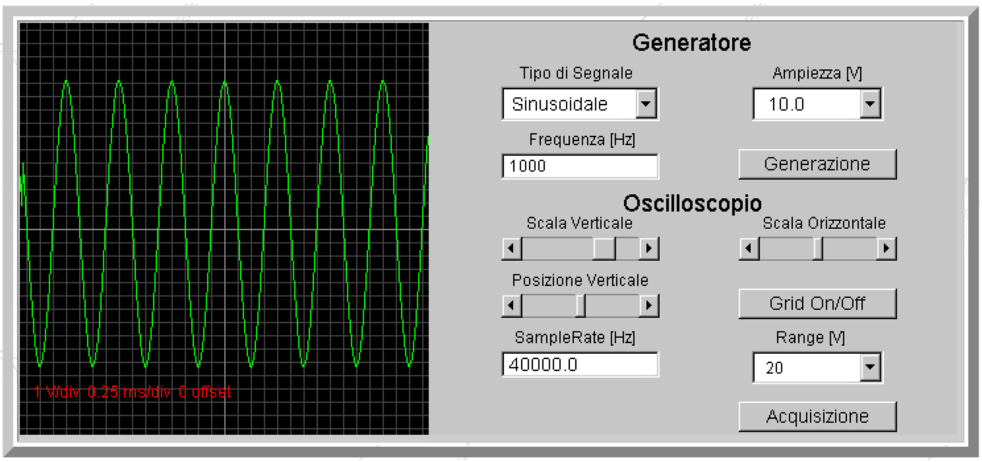

Fig. 2. Front panel for the waveform generation/acquisition VI.

3.3 Remote programming of FIR filters

A very interesting and important component that is widely used in measurement systems is the digital signal processor (DSP). The features such as the capability (i) of implementing complex measurement algorithms, and (ii) to work under strict real-time constrains made it attractive for both research and industrial applications. In accordance with the virtual instrumentation concept it is possible to hide the use of the DSP from the end-user, providing him/her all the

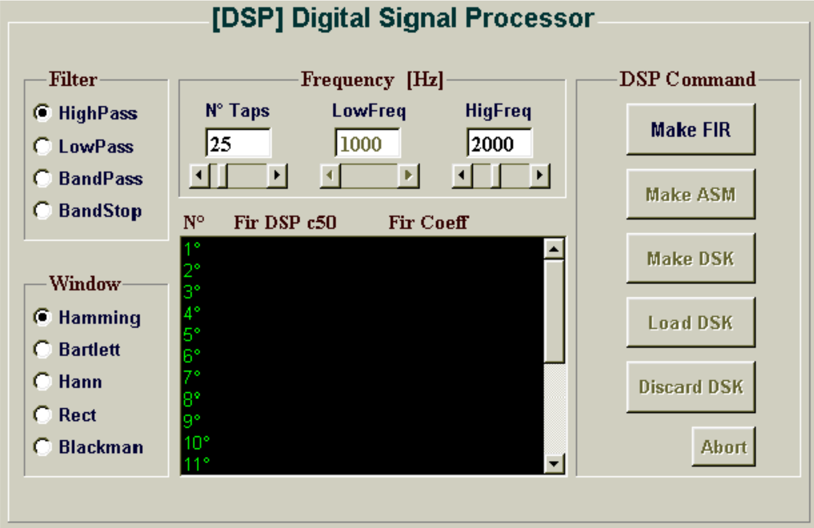

Fig. 3. User interface for FIR filter design.

results through a simple graphical interface. In particular, connecting to the server a Texas Instruments TMS320C50 DSP, mounted on an evaluation board, permitted to set up a useful experiment. The GUI shown in Fig.3 is realised in Java and hides the problems regarding the DSP programming, giving, in the meantime, to the user, the possibility of experiment the FIR filter design.

The system allows designing and testing the users' filter by inputting the filter characteristics, like the number of taps, the type of filter, the high and low cut frequencies, the required window. Then, it guides the user through the DSP programming phases, by producing the assembly file corresponding to the designed filter and loading it into the DSP through an RS-232 interface.

3.4 Remote control of an industrial process emulator

An evolution of the VI concept is the remote control and monitoring of measurement instrumentation [1-3]. Even if the interaction between the user and the instruments takes place through the network, the main idea is still valid: the end user is supplied with a graphical interface that makes the use of the instrumentation easier and friendlier. The final and more challenging step is to realise a distributed measurement system and to makes it to seem to the end-user just like a VI. Fig.4 shows the extension of this concept to an entire industrial process monitoring system. In particular, a self-built emulator of industrial process has been realised. Then, a number of sensors and actuators has been mounted in order to remotely control and monitor it. More in detail, the self-built system emulates a process characterised by the circulation and temperature changing of a fluid, that is water in this case. The fluid level and temperature are continuously monitored by means of a thermocouple and an ultrasonic level transducer. Acting on three electro-valves and a boiler it is possible to change their values. The monitoring signals are wired to the analogue inputs of a National Instruments PCI6023 data acquisition board, while the actuating signals are retrieved from a relay board connected to the digital outputs of the same PCI6023. The whole system can be remotely controlled through the Java user interface reported in Fig.4, so the remote laboratory user can deal with the problems arising in monitoring the industrial processes. The emulator is being currently using for research purposes in the field of real-time remote monitoring [35].

4. CONCLUSIONS

The paper shows the features and the potentialities of a remote laboratory for remote experiments with educational aims. It has been realised within the LESIM of the University of Sannio and already provides some useful experiments to the students of electronic measurements (http://lesim1.ing.unisannio.it). The ongoing activities of the quoted project are devoted (i) to improve the laboratory features by improving the number of available experiments, (ii) to use its architecture for research on distributed measurement systems, with a particular attention to the Quality of Service problems related to them, and (iii) to extend the laboratory to more Universities in order to both provide more experiments and to reach more students.

In order to achieve such targets, new technologies will be examined and eventually exploited to provide suitable tools for a more rapid development of Web accessible VIs not tied to proprietary software. In particular, XML and SOAP seem to be suitable to be used with this aim.

REFERENCES

[1] P.Arpaia, A.Baccigalupi, F.Cennamo, P.Daponte, "A distributed measurement laboratory on geographic network", Proc. of 8th IMEKO Int. Symp. on "New Measurement and Calibration Methods of Electrical Quantities and Instruments", Budapest, Hungary, 1996, pp. 294-297.

[2] P.Arpaia, A.Baccigalupi, F.Cennamo, P.Daponte, "A remote measurement laboratory for educational experiments", Measurement, vol.21, No. 4, 1997, pp. 157-169.

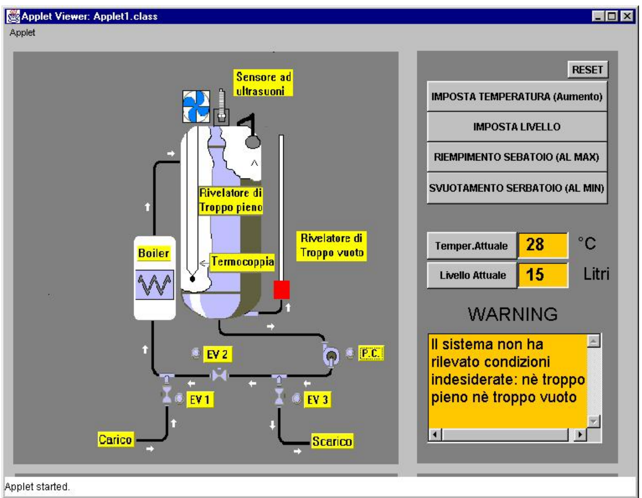

Fig. 4. Graphical user interface for the industrial process emulator.

[3] P. Arpaia, A. Baccigalupi, F. Cennamo, P. Daponte, "A measurement laboratory on geographic network for remote test experiments", IEEE Trans. on Instrumentation and Measurement, Vol. 49, No.5, 2000, pp. 992-997.
[4] D.L.Gaissert, "A Java based test program set development environment", Proc. of IEEE AUTOTESTCON 1999, San Antonio, TX, USA, pp. 775-785.
[5] D.F.Tyler, "Java based automated test systems: management considerations for an open architecture for test", Proc. of IEEE AUTOTESTCON 1999, San Antonio, TX, USA, pp. 699-706.
[6] G.Fortino, D.Grimaldi, L.Nigro, "An agent based measurement laboratory over Internet", Proc. of IEEE AUTOTESTCON 1999, San Antonio, TX, USA, pp. 61-71.
[7] M.Cobby, D.Nicol, T.S.Durrani, W.A.Sandham, "Teaching electronic engineering via the World Wide Web", Proc. of IEE Colloquium "Computer Based Learning in Electronic Education", London, U.K., 1995, pp. 7/1-11.
[8] M.A.Pine, L.A.Ostendorf, "Influencing earth system science education: NASA's approach", Proc. of 1995 Int. Geoscience and Remote Sensing Symp., IGARSS'95, Florence, Italy, vol. I, 1995, pp. 567-569.
[9] J.A.Botti, R.Myers, "Exploring the environment: a problem-based approach to learning about global change", Proc. of 1995 Int. Geoscience and Remote Sensing Symp., IGARSS'95, Florence, Italy, vol. I, 1995, pp.391-393.
[10] D.R.Johnson, M.Ruzek, M.Kaib, "Earth system science education: an Internet-based electronic curriculum", Proc. of 1995 Int. Geoscience and Remote Sensing Symp., IGARSS'95, Florence, Italy, vol. I, 1995, pp.572-574.
[11] E.D. Chesmore, "The teaching of environmental subjects in electronic engineering degree courses", Proc. of the 1993 IEEE Int. Symp. on Electronics and the Environment, Arlington, VA, U.S.A., 1993, pp.79-83.
[12] G.C. Orsak, D.M. Etter, "Connecting the engineer to the 21st Century through virtual teaching", IEEE Trans. on Education, vol. 39, No. 2, 1996, pp. 165-172.
[13] S.M. Blanchard, S.A. Haie, "Using the World Wide Web to teach biological engineering", Proc. of Frontiers in Education 1995 - 25th Annual Conf. Eng. Education for the 2lst Century, Atlanta, GA, USA, vol. 2, 1995, pp. 4c5.9-14.
[14] J. Hughes, J. Borrego, M. Pajaro, "Use of computer graphics for enhancing microelectronics courses in satellite video programs", Proc. of the Tenth Biennial University/Government/Industry Microelectronics Symp., Research Triangle Park, NC, U.S.A., 1993, pp. 197-202.
[15] http://www.techonline.com.
[16] http://lesim1.ing.unisannio.it.
[17] http://www.uninettuno.it
[18] http://www.educatorscorner.com/experiments/index.shtml
[19] A. Baccigalupi, M. Savastano, "Simulation of an electrical measurement laboratory", Proc. of 11th IMACS World Congress on System Simulation and Scientific Computation, Oslo, Norway, vol. 5, 1985, pp. 101-104.
[20] A. Baccigalupi, M. Savastano, "Computer-aided electrical measurement experiments", Proc. of 5th IMEKO Int. Symp. on Intelligent Measurements, Jena, Germany, vol. 2, 1986, pp. 218-220.
[21] M. D'Apuzzo, M. Savastano, "Computerised education in electrical engineering", Proc. of 5th IMEKO Int. Symp. on Intelligent Measurements, Jena, Germany, vol. 2, 1986, pp. 89-92.
[22] L. Benetazzo, M. Bertocco, F. Ferraris, A. Ferrero, C. Offelli, M. Parvis, V. Piuri, "A Web-based distributed virtual educational laboratory", IEEE Trans. on Instrumentation and Measurement, Vol. 42, No. 2, 2000, pp.349-356.
[23] J. M. Thiriet, M. Robert, M.J. Martins, M. Hoffmann, "Pedagogical resources reachable via Internet for teaching intelligent instruments: developments within a European thematic network", Proc. of IEEE IMTC 2002, Anchorage, AK, USA, 2002, pp. 1475-1479.
[24] K. Mallalieu, R. Arieatas, D.S. O'Brien, "An inexpensive PC-based laboratory configuration for teaching electronic instrumentation", IEEE Trans. on Education, vol. 37, No. 1, 1994, pp. 91-96.
[25] L. Finkelstein, "Measurement and instrumentation science - an analytical review", Measurement, vol. 14, No. 1, 1991, pp. 3-14.
[26] S.S. Awad, M.W. Corless, "An undergraduate digital signal processing lab as an example of integrated teaching", Proc. of "Instrum. and Meas. Tech. Conf." IEEE IMTC-97, Ottawa, Canada, 1997, pp. 1314-1319.
[27] Institution of Electrical Engineers, "Sensors and Instrumentation Systems - What Should we Teach? How Should We Teach?" Colloquium organised by the Computing and Control Division, Digest n.96/127, London, U.K., 1996.
[28] P. Arpaia, E. Cennamo, P. Daponte, M. Savastano, "A distributed laboratory based on object-oriented systems", Measurement, vol. 19, No. 3-4,1996, pp. 207-215.
[29] P. Daponte, D. Grimaldi, L. Nigro, E. Pupo, "Distributed measurement systems: an object-oriented architecture case study", Computer Standard and Interfaces, vol. 18, No. 5, 1996, pp. 383-395.
[30] G. Fortino, D. Grimaldi, L. Nigro, "Distributed measurement patterns using Java and Web tools", Proc. of IEEE Autotestcon '97, 1997, pp. 624-628.
[31] D.Grimaldi, M.Marinov, "Distributed measurement systems", Measurement, vol.30, 2001, pp. 279-287.
[32] P.Arpaia, P.Daponte, M.Marinov, "Measurement systems distributed on geographical networks for educational purposes", Proc. of 2000 IEEE International Workshop on Virtual and Intelligent Measurement Systems, Annapolis, MD, USA, 2000, pp. 62-68.
[33] M.Karkowski, W.Winiecki, "A new Java-based software environment for distributed measurement systems", Proc. of IEEE IMTC 2001, Budapest, Hungary, 2001, pp. 397-402.
[34] S. Zimmerman, "Building an Intranet with Windows NT4", Sams-Net Pub., Indianapolis, U.S.A., 1996.
[35] D.Grimaldi, M. Marinov, S. Rapuano, "Performance analysis of RSVP based distributed measurement systems", Proc. of 11th IMEKO Symp. on Trends in Electrical Measurement and Instrumentation, Lisboa, Portugal, vol. I, 2001, pp.243-247.

Authors:
Prof. Gerardo Canfora, Dipartimento di Ingegneria, Università del Sannio, piazza Roma, 82100 Benevento, Italy, Ph.: +39 0824 305804, Fax: +39 305840 E-mail: canfora@unisannio.it.
Prof. Pasquale Daponte, Dipartimento di Ingegneria, Università del Sannio, piazza Roma, 82100 Benevento, Italy, Ph.: +39 0824 305817, Fax: +39 305840, E-mail: daponte@unisannio.it, http://lesim1.ing.unisannio.it
Ing. Sergio Rapuano, Dipartimento di Ingegneria, Università del Sannio, piazza Roma, 82100 Benevento, Italy, Ph.: +39 0824 305817, Fax: +39 305840, E-mail: rapuano@unisannio.it, http://lesim1.ing.unisannio.it