

# A WEB-BASED SENSOR NETWORK FOR DISTRIBUTED MEASUREMENTS

*Aldo Baccigalupi (1), Andrea Bernieri (2), Domenico Capriglione (3)*

<sup>(1)</sup> Department of Computer Science, University of Napoli "Federico II", Napoli, I-80123 Italy  
Phone ++39 (081) 7683170 Fax ++39 (081) 7683167 e-mail: baccigal@unina.it

<sup>(2)</sup> Department of Automation, Electromagnetism, Information Engineering and Industrial Mathematics,  
University of Cassino, Cassino (FR), I-03043 Italy  
Phone ++39 (0776) 299671 Fax +39 (0776) 299707 e-mail: bernieri@unicas.it

<sup>(3)</sup> Department of Industrial Engineering, University of Cassino, Cassino (FR), I-03043 Italy  
Phone ++39 (0776) 299703 Fax +39 (0776) 299707 e-mail: capriglione@unicas.it

**Abstract** – *The paper describes the architecture of a distributed web-based measurement system able to disseminate measurement data over the Internet. After drawing the general architecture here proposed, the paper goes on describing, in somewhat detail, (i) the design choices adopted, (ii) the solutions implemented in order to correctly manage the measurement equipment also in presence of multiple connections, (iii) the assessment for maintaining a high degree of both generality and portability. Finally, a first application concerning a measurement station for the dissemination of environmental quantities over Internet is reported, in order to highlight the capabilities of the proposed architecture.*

**Keywords** – Distributed Laboratory, Remote Measurements, Sensor Network, Environmental Measurements.

## 1. INTRODUCTION

The present work is part of a major project aimed to build up a complex measurement laboratory, distributed on the national area, to be interactively accessed by remote users through the Internet. In this framework, the paper describes the system architecture realised for acquiring measurement data on the field and for making them available to any user connected on the World Wide Web.

The main problems affecting distributed measurement applications concern the management of concurrence coming from the possibility that remote users require to access the same measurement equipment at the same time.

Different solutions have already been proposed in literature [1]-[8]. In this paper, a new approach is proposed aimed to keep separated the problems related to the measurement itself from those concerning the communication management. This approach, once the communication kernel has been set-up, allows the development of distributed measurement applications without requiring a specific communication skill.

In the following, after the general description of the proposed architecture, the main details of the realised measurement and communication system are reported. In

order to test the capabilities of the proposed architecture, a sample application has been developed. It concerns a web-based sensor network for broadcasting environmental measurements realised with commercial hardware and using a non-proprietary development tools.

## 2. THE PROPOSED ARCHITECTURE

The proposed architecture allows three application classes to be developed, characterised by different level of interactivity permitted to the remote user. Specifically:

- (i) *only-viewers (OV)* applications, in which remote users may receive the measurement data, obtained from a predefined measurement task, without any possibility of interaction; typical examples of this class are monitoring applications, meteorological and environmental data dissemination, etc.
- (ii) *single-controller/multi-viewers (SC/MV)* applications, in which several remote users view the data related to a measurement session controlled by a single remote supervisor; possible applications are tutorial experiences in which a tutor/teacher acts as supervisor and the students/apprentices act as remote viewers.
- (iii) *multi-controllers/multi-viewers (MC/MV)* applications, in which a number of remote users can interact concurrently with the application, sharing a number of resources; typical examples of this class are laboratory experiences in which any remote user (student/apprentice) is allowed to train on a specific experimental activity.

In order to achieve this purpose, a flexible and re-configurable architecture has been developed. From a structural point-of-view, it consists in a measurement and communication system organized in two separate layers:

- (i) instrumentation layer;
- (ii) communication layer.

This choice made it possible to develop different measurement applications modifying the *instrumentation layer* solutions, without changing the *communication layer*.

### 2.1 Instrumentation layer

The *instrumentation layer* is devoted to control and execute the measurement operation. The standard measurement hardware (stand-alone instruments, PC-bus Data Acquisition Boards, Field Bus controllers, Smart Sensors, etc.) has been configured to execute the specific measurement session without taking care of the operation performed by the communication layer. A *measurement controller (MC)* has been used, in order to manage the configuration and measurement operations through standard interfaces (IEEE 488, RS-232, VXI-bus, Field Bus, PC-Bus, etc.). From the software point-of-view, it is possible to adopt any of the solutions typically applied in measurement laboratories, which can be developed both in general purpose languages (VBasic, C++, etc.) and in dedicated environments (LabView, LabWindows/CVI, MatLab, etc.).

### 2.2 Communication layer

The *communication layer* manages all the communication devices and protocols necessary to correctly perform the data exchange between remote users and the MC. Therefore, one of the main tasks of the *communication layer* is to permit each one of the aforementioned application classes to be executed taking care of any concurrence problem connected with the specific application. Furthermore, in consequence of the nature of the connection existing between the remote user and the MC (linked directly, to LAN, to WAN or to Internet), the required devices and protocols have to be adopted, in compliance with the ISO/OSI standard. In all cases, a Data Exchange Controller (DEC) has been used, in order to manage the communication from/to the MC. Great significance holds the communication software, which must assure a high performance in terms of communication rate, connectivity and load independency. From the remote user point-of-view, it is more advantageous that the software tools, necessary to access the whole measurement system, are quite simple and already available on the remote computer. To this aim, a *client/server web-based structure* has been adopted; in this way, a simple *client web-browser* (e.g., Netscape, Internet Explorer, Mosaic, etc.) usually available in the standard operating systems can be used, avoiding the need to distribute *ad-hoc* software. In particular, a suitable *web-server* has been built and used to publish web pages including measurement data and control procedures.

In Fig. 1, the general scheme of the proposed architecture is sketched.

## 3. AN APPLICATION EXAMPLE

In order to test the proposed architecture, a first example is reported in the following. It regards an *OV* application, devoted to disseminate, via Internet, the measurement data collected by a meteorological station located at the Faculty of Engineering of the University of Cassino.

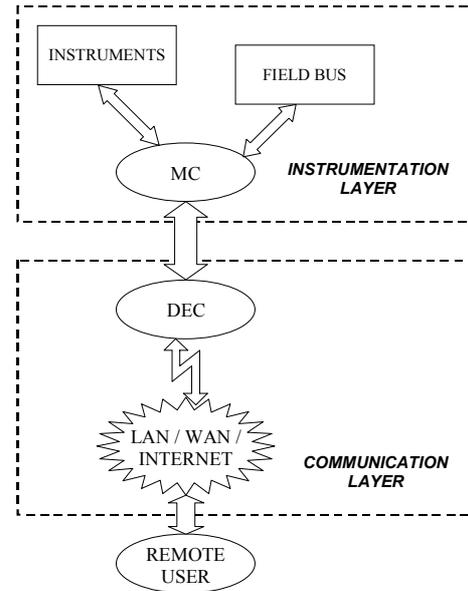


Fig. 1. Structure of the proposed measurement & communication system.

### 3.1 Hardware architecture

The architecture of the measurement system is reported in Fig. 2. Three sensors of environmental quantities (a thermo-hygrometer, a barometer and an anemometer) with normalized outputs (0-5V or 4-20mA) have been used and mounted on a suitable pole. The signals are collected using a Field Bus acquisition module (National Instruments FP A1110); it is controlled by another Field Bus module (National Instruments FP1000), equipped with a standard RS-232 interface; a third module (National Instruments ENET-232/2) links, via RS-232C, the Field Bus controller to an Ethernet network.

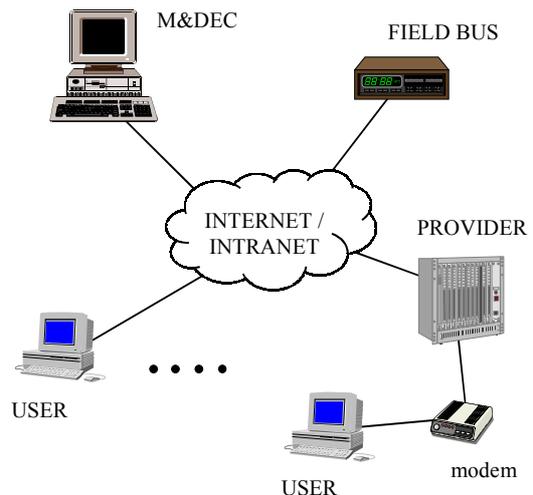


Fig. 2. Architecture of the distributed measurement system.

The whole measurement system is accessed from the MC using a univocal IP address, which is stored in the ENET module during the configuration phase. In this way, the MC

could be placed in whatever location, only equipped with an Internet Protocol network link.

Using these communication devices, the whole Field Bus system appears to be connected to the MC by means of a standard RS 232C additional port, without any reference to the physical Ethernet link or to the Internet Protocol.

For the particular developed application, an hardware simplification has been adopted. In fact, owing to the type of measurement application and to the limited number of measured quantities, it is possible to load both the tasks assigned to the *measurement* and *data exchange controllers* in a single computing system (*M&DEC*). To this aim, a Pentium III (800 MHz 256 MB RAM) computer has been used, equipped with a standard Ethernet (10/100 Mbps RJ45) card. A Microsoft Windows 2000 Operating System has been chosen, in order to assure both the MC operation and, with the bundled Internet Information Server, the DEC functions.

### 3.2 Software architecture

The software is organised taking into account the logical structure of the hardware architecture. In particular, three different procedures were developed:

- (i) *measurement procedure*, running on the MC;
- (ii) *data exchange procedure*, running on the DEC;
- (iii) *remote user procedure*, running on the remote computer.

(i) The *measurement procedure* controls and polls the measurement hardware. Borland C++ language has been used to build a program able to send commands to the Field Bus modules. This choice was adopted for assuring a high degree of generalization and portability along with a good performance in terms of execution time. For this reason, proprietary environments (like LabView, MatLab, and so on) have been avoided.

(ii) The *data exchange procedure* is devoted to perform a link between the measurement procedure and the remote user procedure. To this aim, multiple tasks are necessary, in order to assure the correct information flow. In particular, it is necessary to perform: (a) the reception and management of the connection request from one or multiple remote users; (b) the management of a measurement session on the MC; (c) the data exchange between the MC and DEC; (d) the sending of measurement data to the remote users. Also in this case, the whole data exchange procedure has been written in standard Borland C++ language, in order to maximise the computing performance and simplify the interconnections with the measurement procedure.

(iii) The *remote user procedure* is devoted to make available all the measurement data to the remote user. To do this in a web-based environment, two choices are possible: (a) the use of a standard web browser which access to HTML pages on the web-server, and (b) the use of a standard web browser which loads and locally executes a suitable *applet* (able to directly read the measurement data present on the MC) from the web-server. The second solution is quite

suitable from a measurement point-of-view, since it permits to directly read the measurement data without the interposition of other HTML structures (forms, tables, etc.) which would require a temporary storage on the web server. In this way, the better performances in terms of response time are reached. Moreover, in the case of modifying the *remote user procedure*, it is necessary to bring up-to-date only the applet available on the web server: in the next communication session, the remote computer will load the new version automatically. The adopted applet was written in standard Java language, usually supported by the most common web browsers.

Fig. 3 shows the flow chart of the whole software procedure, with the interactions between the different hardware/software sections. In particular:

- (i) at the remote user connection request, the web server sends the applet to the remote client browser;
- (ii) the DEC creates a socket connection with the remote user;
- (iii) the DEC includes the remote user IP address in a suitable user's array, in order to manage all the concurrent connections;
- (iv) the DEC creates a socket connection with the MC;
- (v) the MC initialises (if necessary) and polls the measurement instruments, receiving the measurement data;
- (vi) the MC sends the measurement data to the DEC through the socket connection created in step (iv);

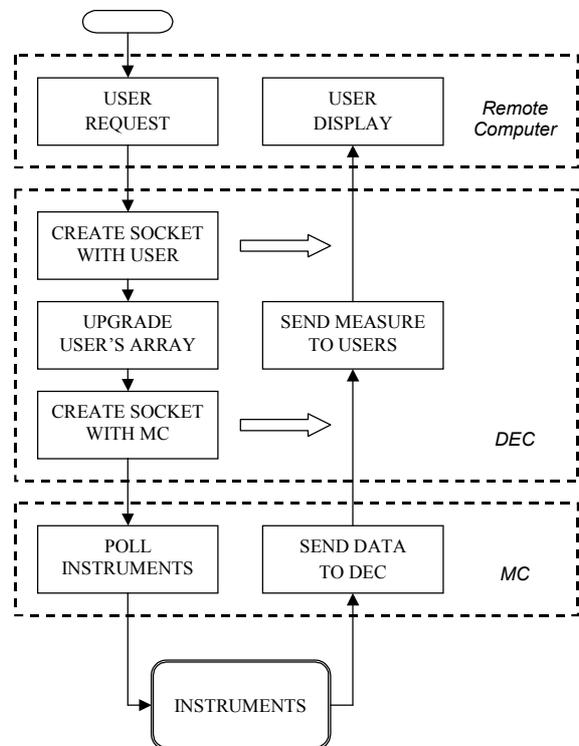


Fig. 3. Flow chart of the software measurement & communication procedure.

- (vii) the DEC sends the measurement data to the remote user browser through the socket connection created in step (ii);
- (viii) the applet displays the measurement data on the remote user computer.

Concurrent remote requests have been managed through a proper user's array. In fact, through the sockets described in step (ii), one for each user, it is possible to send simultaneously the measurement data to all the users previously loaded in the array. Moreover, the management of the user's array, performed by the DEC, allows the connection status control to be performed, avoiding connection errors in the case of an user disconnection.

It is worth noting that the use of the above mentioned sockets allows the data transmission from the measurement system to the remote computer to be performed directly, so avoiding the necessity of intermediate operations (i.e. web-page rewriting and republishing).

Fig. 4 shows the user display, generated by the applet, with the measurement data arranged in an agreeable form. The data refresh depends both on the data acquisition time of the measurement system and on the data transfer time between the sockets through the network connection. At the moment, the refresh time is limited only by the data acquisition time, fixed at 1s, since the data transfer time is negligible thanks to the low number of measurement data involved.

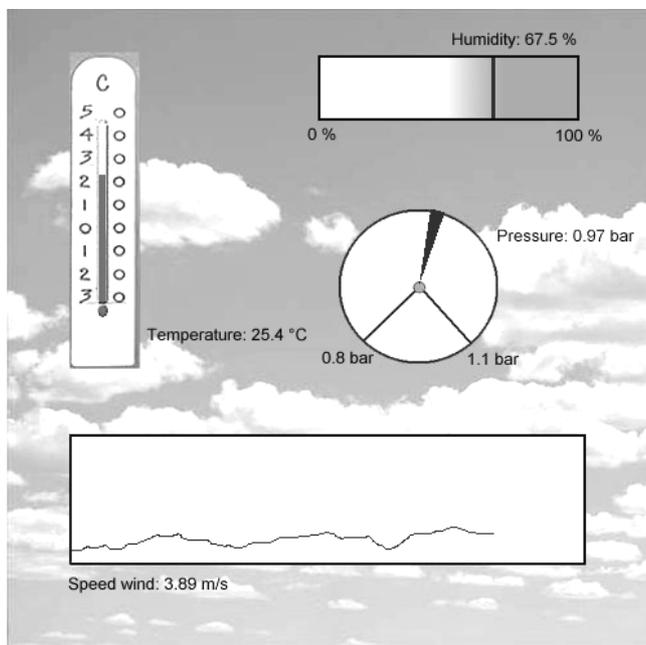


Fig. 4. User display of the meteorological quantities.

#### 4. CONCLUDING REMARKS

A structured architecture for distributed measurements has been presented and discussed. The solution to split the

whole structure in two separate layers (*instrumentation and communication layers*) allowed different measurement applications to be easily realised only modifying the *instrumentation layer* architecture. At the same time, the proposed architecture made it possible to develop measurement applications characterised by different levels of interaction between the measurement devices and the remote users (*only-viewers, single-controller/multi-viewers, multi-controllers/multi-viewers*) without taking care of communication management.

In order to increase the generality and portability of the proposed architecture, a client-server web-based solution was adopted. In this way, a simple web browser is needed for each remote-connected user. Moreover, the choice to adopt a client procedure based on a Java applet allows both an easy upgrade of the procedure itself and a fast data communication to be realised.

A first example, a simple *only-viewers* application, is presented to test the proposed architecture. It is related to disseminate on the Internet meteorological data acquired by a Field-Bus-based network of environmental sensors.

Further research activities are still in progress. They are aimed to realise applications belonging to the other mentioned classes and to test the correct working of the communication layer also with more complicated applications involving a higher degree of interactivity. Problems related to the data transmission rate for on-line measurement applications are still under investigation.

#### ACKNOWLEDGMENTS

The authors wish to thank Dr. Gianluigi Fiorillo for his valuable support in the measurement and communication system realisation.

#### REFERENCES

- [1] Blanchard S.M., Hale S.A.: *Using the World Wide Web to teach biological engineering*. Proc of Frontiers in education 1995 – 25<sup>th</sup> Annual Conference Eng. Education for the 25<sup>th</sup> Century. Atlanta GA – USA.
- [2] Algeria F., Ramos H.G.: *A remote controlled automated measurement system*. Proc. of Instr and Meas. Tech. Conf. IEEE IMTC 97. Ottawa May 1997.
- [3] Bertocco M., Ferraris F., Offelli C., Parvis M.: *Training of programmable instrumentation: a student laboratory*. Proc. of XIV IMEKO World Congress June 1997 – Tampere (FIN).
- [4] Arpaia P., Cennamo F., Daponte P., Savastano M.: *A distributed laboratory based on object oriented systems*. Measurement 19 (3-4), Nov-Dic 1996.
- [5] Arpaia P., Baccigalupi A., Cennamo F., Daponte P.: *A distributed measurement laboratory based on geographic network*. Proc. of 8<sup>th</sup> IMEKO Symp. - Sept. 1996. Budapest (Hungary)
- [6] Arpaia P., Baccigalupi A., Cennamo F., Daponte P.: *Ethernet application perspectives for distributed measurement systems*. Proc. of XIV IMEKO World Congress June 1997 – Tampere (FIN).
- [7] Arpaia P., Baccigalupi A., Cennamo F., D'Apuzzo M.: *Remote characterization of scan conversion based transient digitizers*. Proc. of Instr and Meas. Tech. Conf. IEEE IMTC 97. Ottawa, May 1997.
- [8] Lukaszewski R., Bobinski P., Winieki W.: *Java based Distributed IEEE-488 Measuring System*. Proc. of XVI IMEKO World Congress – Sept. 2000 Wien (Austria).