XVI IMEKO World Congress
Measurement - Supports Science - Improves Technology - Protects Environment ... and Provides Employment - Now and in the Future
Vienna, AUSTRIA, 2000, September 25-28

JAVA-BASED DISTRIBUTED IEEE-488 MEASURING SYSTEMS

R. Lukaszewski, P. Bobinski and W. Winiecki
Institute of Radioelectronics
Faculty of Electronics and Information Technologies
Warsaw University of Technology, 00-665 Warsaw, Poland

Abstract: A new concept of distributed IEEE-488 measuring systems using Java architecture is proposed. A connection of external modules (standard library for National Instruments IEEE-488.2 interface board) to source code in Java is described. Next, an example of the system, that uses a communication based on sockets for remote controlling of instruments, also via Internet, is presented.
Keywords: distributed measuring system, IEEE-488, Java

1 INTRODUCTION

Fast development of computer networks, that link personal computers, creates new possibilities of its application in measuring systems. Development of information technologies leads up to appear advanced network services, such as publishing various information (WWW) or remote operating of application. One of the possibilities of applying new networks technologies in metrology is remote access to advanced metrological services, including specialised processing procedures, unique measuring instruments, access to novel measuring systems and to research laboratories. The usage of distributed networks enables one to simultaneous applying above elements by many users independently from their place of stay.

In the paper a concept of virtual instruments system is presented, as a part of Virtual Laboratory in Internet, that is realised as a Dean's grant. Main deal of the described work is to ensure remote controlling of IEEE-488 instrument via distributed network, in particular: designing program interfaces and dialogue interfaces for virtual instruments system.

2 THE NEW CONCEPT

There is many methods to ensure the user remote access to IEEE-488 instruments [8] from terminal that is connected to LAN/WAN network. All these methods are based on architecture Client-Server using standard network protocols (for example: TCP, IPX, UDP). Most of the solutions [4], [5], [6], [7] are based on integrated environments as LabView, LabWindows or HPVEE, that have built-in procedures that ensure an access to network transmission protocols. Easy and fast designing of complex graphical user interfaces (GUI) is a great advantage of these solutions. Their basic disadvantage is a fact, that client program, created in this way, is a separate application and must be sent to user computer, installed there and run on. Another solutions [1], [2], [3] use WWW mechanism and enable one to control remotely a measuring system from WWW page using standard Internet browsers (as: Netscape Navigator, Internet Explorer). Such an access to available services is very easy for users, so we decided to apply Java architecture [10] in the project and we have taken an assumption, that remote control application will be accessible as an applet via WWW server. Data flow diagram is presented in Figure 1.

3 IMPLEMENTATION

There were taken the following assumptions to realise the software:

- an mediator role between measurement application (applet) and GPIB function library is held by TCP/IP server (called GPIB server) that interprets (coming on an appropriate communication port) commands and calls board driver functions,
- a Java class for communication between an applet and the server is created,
- a special protocol for communication between the GPIB server and an applet (strictly: a class, that is used by the applet) is applied,
- the applet with the above class have to be available from WWW page using HTTP protocol.

Designed software was divided into four functional modules:

- data source interface,
- data source server,
- client interface,
- server WWW.

As a data source (Fig. 2), measurement instruments with IEEE 488.2 interfaces are used. Data source interface consists of IEEE-488 bus, IEEE-488.2 controller card (National Instruments), and also software drivers and libraries supported by producers.

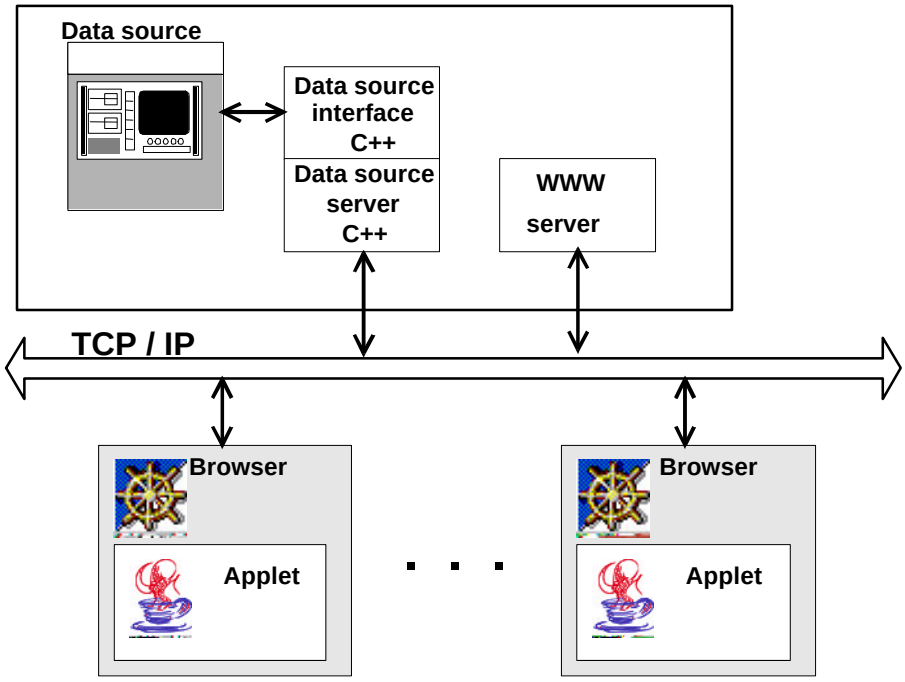

Figure 1. Data flow diagram in Virtual Laboratory.

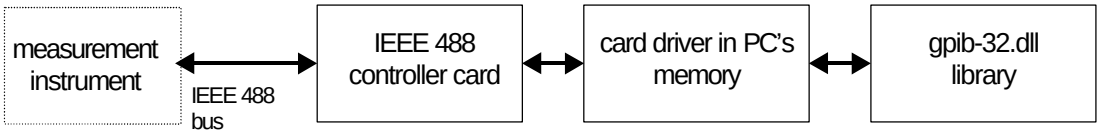

Figure 2. Block diagram of data source interface.

Communication between the applet and IEEE-488 card is performed using the software driver supported by the producer. The software driver is loaded to computer memory and executed, when the operation system MS Windows is started. The IEEE-488 card is treated by the computer as the system device. Measurement application can calls the card driver using an appropriate libraries with interface functions. The 32 bit dynamic library (gpib-32.dll) is preferred in the MS Windows.

Data source server (called as GPIB server), created in Visual C++ 5.0, is based on data transmission via sockets (TCP/IP protocol) [9], [10] and includes an interpreter of commands, that are coming on appropriate communication port. It takes a function of an mediator between measurement application (applet) and GPIB-function library.

Communication between several elements of the project is presented in Fig.3.

GPIB server is based on the idea of echo-type server, that uses standard WIN'95 WINSOCK library. This library enables one to open and operate a transmission channel, that uses any port. After starting the program, the server is waiting for the connection with the port, until a message from a client appears. Such a message is translated by the interpreter. The interpreter verifies and interprets the message and decides about its correctness and destination. In the most cases, these messages are such as call to the GPIB library.

To build a client interface (Java application or applet) specialised class (GPIB.class) was created, that contains methods corresponding to functions of IEEE 488.2 controller card. Communication between Java applets (or application) and GPIB server is carried out using specially designed for this purpose protocol. The protocol was created to limit transmitted data amount, to limit a possibility of

faults appearance when a measuring system is working, and to easy detection and interpretation of errors.

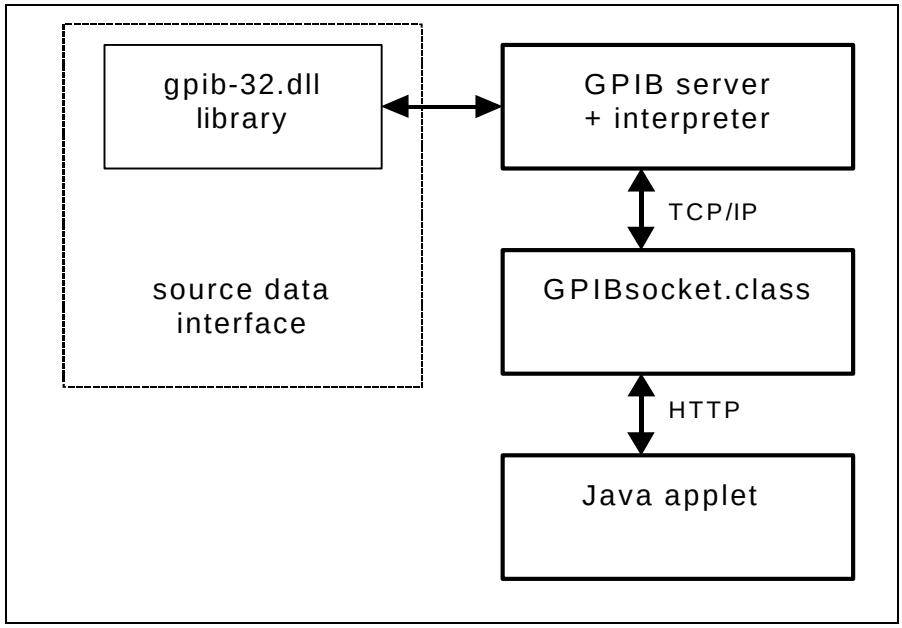

Figure 3. Block diagram of socket based conception.

Particular methods calls appropriate GPIB procedures using GPIB server and the interpreter of control words. Communication between Java applets and GPIB server is performed using specially designed for this purpose protocol. The protocol consists of a set of appropriate, ASCII-based, words. Each word corresponds to definite function, that is equivalent in meaning both for the server and GPIB socket class. These functions can be divided into two groups:

- control functions,
- communication functions.

Communication word consists of two main fields and separators. The first field is a function identificator and consists of two bytes. The second field covers a list of function parameters, which are separated by special characters (see Fig. 4). Communication functions are used to ensure a correctness between a class and the server.

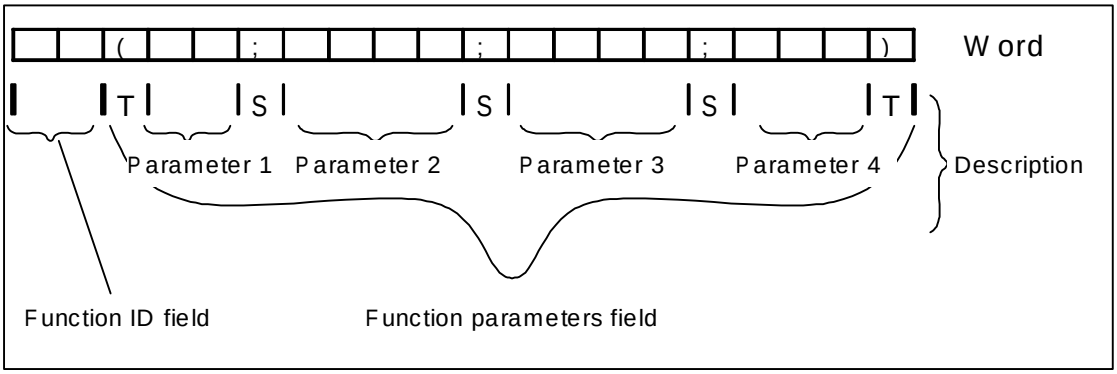

Figure 4. Transmission communication word: T – half word terminator, S – function parameters separator.

A set of words, responsible for calls of appropriate procedures of GPIB interface and for return information from these procedures composes control functions.

Simple WWW server was created in Java to make available (HTTP protocol) client applet with GPIB.class. The server is started in the same computer, as GPIB server and GPIB card.

Diagram of measurement system using wide area network is presented in Fig. 5.

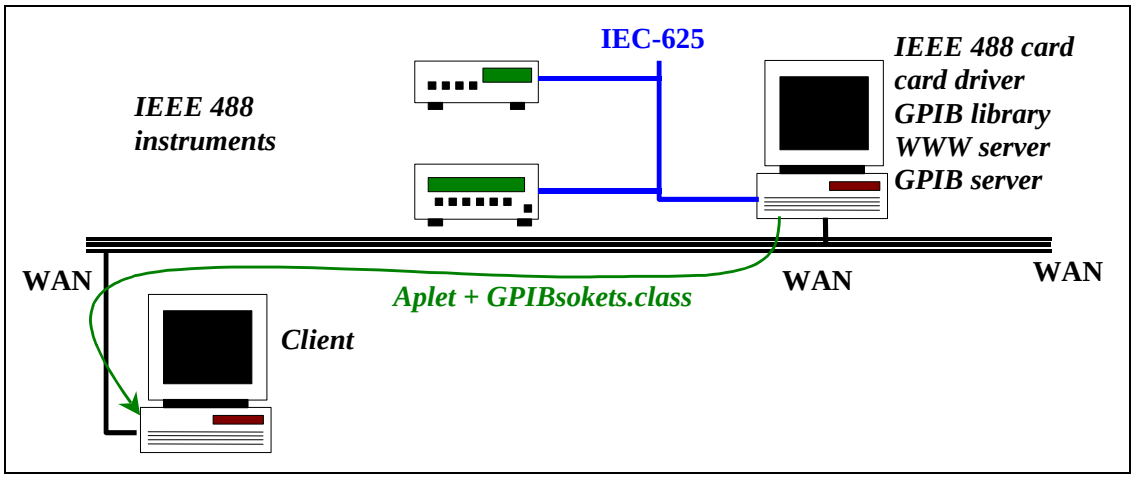

Figure 5. Diagram of measurement system using wide area network.

4 EXAMPLES

Presented concept was a starting point for building measurement applications. The applications, accessible using standard browsers, were built as the applets. Panels of virtual instruments, based on generator HP33120A and multimeter HP34401A, were designed. Student laboratory exercise for learning SCPI language was created. All these applets use directly GPIB.class methods.

The graphical user interface, the Java applet, is built using event driven based programming technique. Most of the graphic user interface objects generate events (in the moment of user action, for example: pushing a button), that are serviced by the software. As an example, an applet of virtual multimeter, that is based on multimeter Hewlett Packard HP34401A connected with a computer via IEEE-488 interface, is presented in Fig. 6.

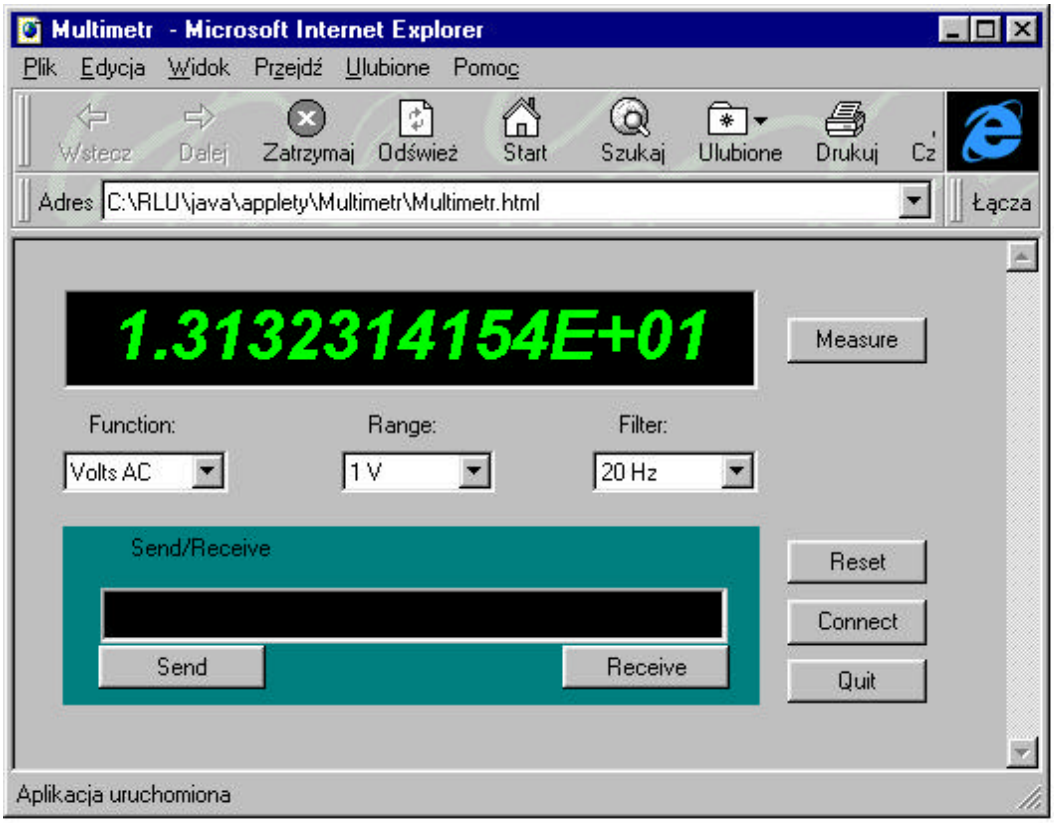

Figure 6. Virtual multimeter front panel.

The virtual multimeter panel covers the following objects:

- push buttons “Connect“ and “Quit“ responsible for performing and stopping connection to server,
- push button „Reset“ responsible for setting the multimeter into the initiate state and for clearing all registers and queues,
- a set of dialogue fields, that enables one to set basic parameters of measurements,
- push button „Measure“, that enables one to start a measure,
- graphic fields for displaying measurement results,
- push buttons “Send“ and “Receive“ responsible for sending programming data to the instrument and reading data from instrument output queue,
- graphic fields for displaying programming data sended to the instrument and data received from it.

5 CONCLUSIONS

As a result of presented research, a set of software tools was prepared, that enable us to realise virtual instruments with IEEE-488.2 interfaces and to control its via distributed network. A set of virtual instruments and a student exercise as a part of Virtual Laboratory in Internet was worked out. These examples have confirm usefulness of proposed concept.

REFERENCES

[1] P. Kadionik, T. Zimmer, Y. Danto, RETWINE: A New Distributed Environment for Microelectro-nics Instrumentation Learning and Measurements, in: *Proc. 16th IEEE IMTC Conference* (Venice, May 24-26, 1999), CD ROM.
[2] C.-P. Young, W.-L. Juang, M.J. Devaney, Real-time Intranet Controlled Virtual Instrument Multiple-circuit Power Monitoring, in: *Proc. 16th IEEE IMTC Conference* (Venice, May 24-26, 1999), CD ROM.
[3] K.B. Lee, Distributed Measurement and Control Based on the IEEE 1451 Smart Transducer Interface Standards, in: *Proc. 16th IEEE IMTC Conference* (Venice, May 24-26, 1999), CD ROM.
[4] L. Benetazzo, M. Bertocco, F. Ferraris, A. Ferrero, C. Offelli, M. Parvis, V. Piuri, A Web-Based Distributed Virtual Educational Laboratory, in: *Proc.16th IMTC Conference* (Venice, May 24-26, 1999), CD ROM.
[5] G. Fortino, L. Nigro, A Multimedia Networking-Based Approach to the Development of Distributed Virtual Instruments, in: *Proc. 16th IEEE IMTC Conference* (Venice, May 24-26, 1999), CD ROM.
[6] M. Bertocco, M. Parvis, An Auto-Routing Multi-Server Architecture for High-Education Training on Instrumentation and Measurement, in: *Proc. 16th IEEE IMTC Conference* (Venice, May 24-26, 1999), CD ROM.
[7] M. Bertocco, F. Ferraris, M. Parvis, A Remote And Distributed Student Laboratory, in: *Proc. IMEKO XV World Congres* (Osaka, June 13-18, 1999), 1999, CD ROM.
[8] NI-488.2 Function Reference Manual for DOS/Windows, NI Corp., Austin, 1995.
[9] http://www.sockets.com/
[10] ttp://www.sun.com/

AUTHORS: Ass. Robert LUKASZEWSKI, M.Sc., Ass. Piotr BOBINSKI, M.Sc. and Ass. Prof. Dr Wieslaw WINIECKI, Institute of Radioelectronics, Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warsaw, Poland, Phone Int. + 22 660 7341, Fax Int. + 22 255248
E-mail: R.Lukaszewski@ire.pw.edu.pl