AN OBJECT-ORIENTED MODEL FOR DISTRIBUTED SYSTEM MONITOR

J. A. Pérez Venzor, D. García, A. Delgado
F. J. Enríquez A., H. M. Loya, R. E. Pérez, M. I. Castellanos

Universidad Autónoma de Ciudad Juárez, Chihuahua, México
Departamento de Ingeniería Eléctrica y Computación
Tel.: (+52) 656–688 48 00, Ext. 4741
E-mail: jorperez@uacj.mx

Abstract: This paper deals with the application of the paradigm Oriented to Objects in a distributed measurement system. The purpose of this study is to obtain a flexible electronic test station model that will allow modifications in the procedures with no modifications to its main code. The model allows the integration of GPIB equipment in a distributed measurement system Via local area network (LAN) or Internet, with the ability to store/read the results of the measurements through a distributed database. The outlined plan shows a local interface for tests and also interfaces for the Client-Server. The model shows a high degree of flexibility to be utilized in different configurations both in equipment distribution and in realization procedures.

Keywords*:* Distributed control, GPIB, DMSs.

1. INTRODUCTION

With the development of personal computers, very important advances in instrumentation have been achieved, this has a significant improvement for software that control and communicates with measurement instruments, thereby creating as a whole, a type of computer controlled virtual instrument. The instrument in general is conformed by a test station implemented in accordance with the particular requirements capable of carrying out a series of specific tests in a certain application within a Distributed Measurement System (DMSs) (figure 1). Among the most common measuring instruments are the ones that are transmitted through a bus named GPIB (General Design Interface Bus) which was designed by Hewlett Packard (1965) originally called HPIB.

Nowadays the development of distributed measurement systems requires better tools in order to be able to transfer, maintain, and modify systems. Currently the modifications to the design of the systems are sometimes measured in weeks and even in months; it is no longer possible to create systems from scratch every time a new instrument o model is incorporated into the project. Moreover, if there are one or several groups of design engineers within the project, they can use different programming languages, including Microsoft Visual Basic, Visual C++ or NI LabVIEW, and through component software technology like ActiveX technology, connect the different stages of the project. It is important that the software components of an integrated measurement system should be easy to maintain and quick to modify.

Below the raising of the problem and its justification are described; furthermore, related studies reported in literature are mentioned. Afterwards, the proposed solution and its repercussion in a distributed measurement system are presented.

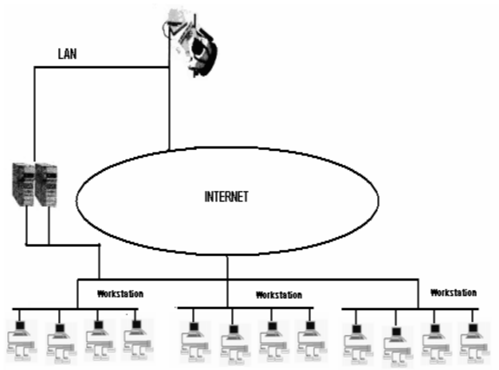

Fig.1. Distributed Measurement System (DMSs)

2. PROBLEM DESCRIPTION

In this article the analysis and design of a flexible and modular OO model of a test station integrated by electronic measurement instruments with GPIB interface is outlined. Currently, a large number of manufacturing companies like the automotive and telecommunications use these types of instruments for the on-line acquisition of quality parameters in the production process. Due to market demands the product is subject to modifications which are more and more frequent; therefore, the processing stations on the production line suffer modifications. In the case of the electronic test stations, these modifications go from a change in the reference value of an electrical variable in the program to adding and/or eliminating measurement instruments from the test system.

It is important for the setting in motion of the production process of a new model of the product, that the test system endures these kinds of quick modifications. At the present time, there are a lot of challenges for engineers and scientists to integrate automated measuring systems. There is a necessity in the market to have new products of better quality and with a very short life cycle. Therefore, researchers face the challenge of rapidly creating automated measurement systems in order to test products during their manufacture process. The latest trends in the integration of manufacturing companies requires the handling of the test station like a subsystem integrated into the production management system; this situation makes the challenge for designers more complex. In the studies reported in literature, flexibility and modulability characteristics are not achieved when the instruments in the system are changed, this is, carrying out modifications in the configuration of the equipment without there being changes in the station's control software, that is to say, without opening the system.

3. BACKGROUND

The research on measuring systems is diverse and comprehensive as there are commercial solutions like the ones offered by sellers like National Instruments and academic solutions. There are studies on automatic test stations like the HONEYWELL H2600, in [1] using GPIB and VXI interfaces for connection with the instruments which produce tests for the electronic cards of F-15 and F-16 airplanes where the test conditions to be carried out are stored in memory; however, no flexibility is shown for carrying out changes in the instruments which make up the station leaving it not very flexible to future expansions in regard to new connections of instruments. In [2] the realization of a distributed laboratory of measurements oriented to objects in which (DMSs) Distributed Measurement Systems applied to an Internet system, being reported, in which the GPIB bus is used with internet protocol and where instruments are assembled like a workstation in a PC.

In [3] a laboratory of virtual instrumentation for apprenticeship being proposed where the instruments are found on a personal computer which in turn is found on a local network; its goal is to attain the best understanding of electronic engineering students concerning the handling of software for communication applications and developing programs. This study is oriented to the structure of the system without considering the reusability of software. [4] Explains an architecture for building a system of interchangeable tests in regard to the bus. The need for handling different buses like: GPIB, VXI, and the IVI interchangeable virtual instruments are raised where all of the instruments of these architectures are combined in a single system and thereby combining the different I/O of each bus. In [5] a system open to an IEEE-488 interface of assigned measurement instruments in an environment for microelectronics being proposed, utilizing UNIX as an operative system platform where the managing interface of the device will be implemented in a remote access of the GPIB Bus; the application of the interconnection will be in a graphic interface and will be applied remotely to an instrument as seen in figure 2; the handling of the GPIB controls are the ones handled by the standard IEEE-488.2 and handled independently to each instrument. In [6] a distributed measurement system based on Java shows where the instruments are handled through the internet, the client/server proposes this system based in object oriented programming and through this client server system, it configures and monitors the measurements of the system.

4. PROPOSED SOLUTION

The Unified Modeling language (UML) is a graphic language which was used to represent the characteristics of the system. UML allows for the filing, communicating, and leveling of knowledge. The models file knowledge (semantics). The architectural views organize knowledge according to directives that express idioms of use, and the diagrams depict the knowledge (syntax) for communication [7].

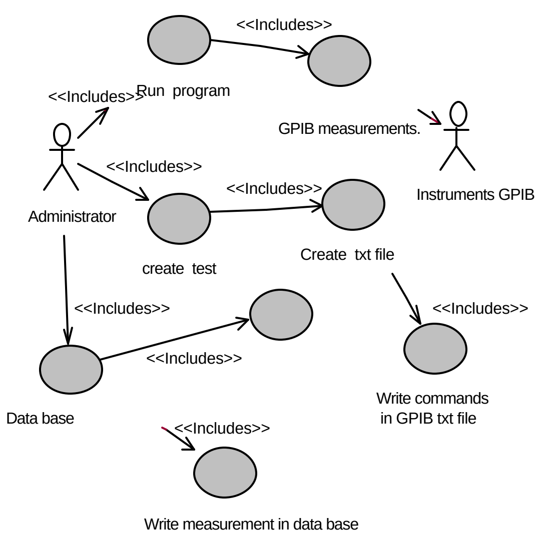

Fig. 2. Diagram case for the system used

A first approximation of the proposed solution is presented in the form of cases of use in figure 2. The main cases of use identified in the system are the following: carrying out a test by means of a PC for equipment that will be capable of using the GPIB bus, creating a new test with some other GPIB instrument, creating a series of commands (program) written on a text file for the system to perform, and storing the obtained results on a database. In figure 3 the diagram of general classes is seen where it can be noted that a test station can contain an indefinite number of personal computers as well as equipment with GPIB interface and a database for storing the information shown by the performed measurements. In figure 4 the classes of the test station computer system are shown. The clsEquipoGPIB class is the central part of the system which is in charge of reading the program to be carried out at the station. The program is inserted by the user in text archive form. Each physical piece of equipment is wrapped by the clsGPIB class, which is in charge of implementing the communication protocol between the equipment and the computer system, specifically with the clsEquipoGPIB class. Each program command (text archive) is sent by the clsEquipoGPIB class to the clsGPIB class. The clsBaseDatos class is in charge of handling the station's database. This class receives the result of the measurements sent by the clsGPIB class through the clsEquipoGPIB class. The activeX GPIB component is composed by the clsGPIB class; this class is very important; the reading and writing of the measurements of the instruments is handled in it; this component is capable of reading/writing on any GPIB instrument without needing to make changes in its code (fig.4).

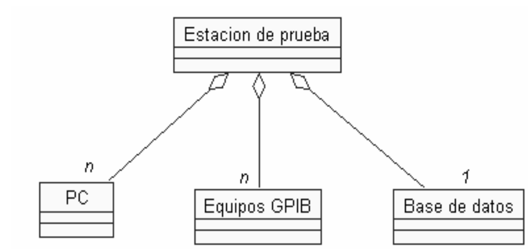

Fig. 3. Diagram general of classes for the system

The commands that this component receives are issued from a text archive. The component that is in charge of reading this archive is ActiveX EquipoGPIB which contains the clsEquipoGPIB class; this class is in charge of sending the commands of the archive to the ActiveX GPIB component in the clsGPIB class for its completion. The EquipoGPIB component is also in charge of receiving the results of the measurements and storing them in the database by means of the clsBaseDatos class.

The inicio component contains a frmVisor form (figure 4) which is in charge of displaying the archives for modifications (editing); it also shows the results of the measurements. This component also offers navigation through the data and printing the results of the measurements. Furthermore, in this component the implementation of the client for performing remote form tests is found; whether it be on an LAN network or intranet, the server is found on the EquipoGPIB component.

The object oriented programming provides a good base for modeling this system where the interaction of software among the different components interacts like independent objects.

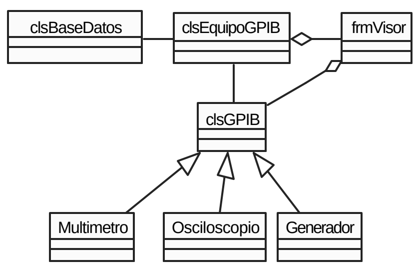

Fig. 4. Diagram classes for the GPIB probe station

Bearing in mind that this association will be the one utilized by the distributed measurement system and the characteristics of the handlers of the EquipoGPIB components to perform the measurements, the system can execute a test by means of a local LAN network or Intranet. With this system one or several measurements can be performed according to one's wishes, utilizing the previously seen components: Inicio component, EquipoGPIB component, and GPIB component. This system can be composed by several subsystems with a central control for them through network technology and software methods. With this way of sending the commands we avoid the protocol of internet connection of the IEEE-488.2 [8].

The characteristics of an interoperable work network and the ease for reusability of the software can be employed in the distribution of distributed test stations. In an intelligent measurement system (IMS) the operator shapes the system to be connected to the instruments as well as running the software modules for each one of them. Each time a test is performed in the distributed measurement system it is necessary that a server and a client exist, the server being one or several work stations with GPIB equipment which are found remotely the client is in charge of distributing to each server the commands that will be sent to each work station for a specified piece of GPIB equipment. The commands will be in a text file in the client; then the file will be sent to the server which requires that test; the server sends a message to the client which says that the file has been received. The way of performing the test, once the text file and its location within the test station is received, follows the steps as previously described.

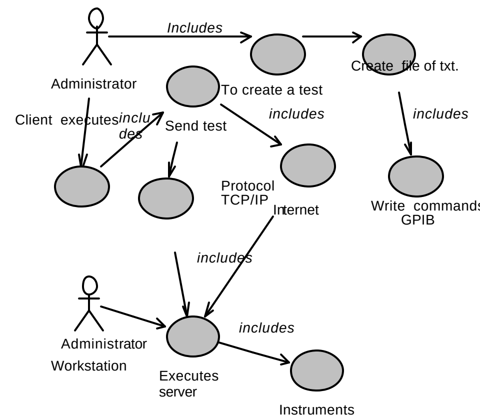

Fig. 5. Cases for distributed system of measurement

In figure 6, the client class is the one which administers the test and is in charge of sending the file to the TCP/IP classes or, if applicable, to a LAN where several servers can exist that will administer up to 15 pieces of GPIB equipment each. In figure 7 the sequence diagram for a test in a server is shown. In this case, for a station where we have the client and the server, the file of a test is selected in the client to, thereby, begin it; this file is immediately sent through a communication protocol whether it be TCP/IP or LAN internal network so that, later, the file gets to the server and is kept at the test station to be performed. The server, once this process is completed, sends a message to the client to inform that the former has completed the process and is ready for the next test.

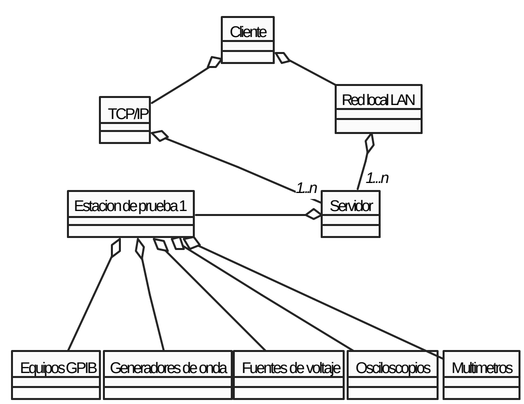

Fig. 6. Diagram of classes for the distributed measurement system

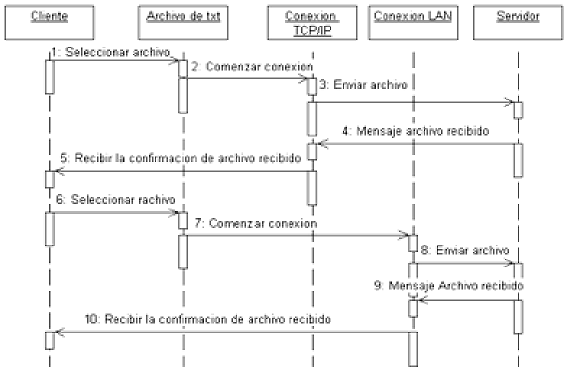

Fig. 7. System Sequence diagram

5. CONCLUSIONS

The result of this project is the proposal of an integrated software platform for automated distributed measurement systems which allows an integrated and modular structure. This platform facilitates the configuration of the test station regarding to instruments, as well as to their handling by means of a test station or through access in a local area network (LAN) or by Internet, performing measurement tests with results stored in a database. This system offers the following benefits: Increase productivity throughout the development, display, maintenance, and modification of processes.

- Obtains a better performance for the test.
- Performs tests of several pieces of equipment in a local area network or by Internet, which allows for its applications in industries, laboratories, and teaching, obtaining a high level of versatility in its use.

The system is capable of integrating several measurement devices, GPIB instruments, to systems of high level production management, like the one carried out through the iMRP plan at the automatic control engineering laboratory of the Instituto Tecnologico de Chihuahua [9] as well as at the Electrical Engineering laboratory of the Universidad Autonoma de Ciudad Juarez, Chihuahua. The distributed measurement system remains open to new improvements in the future thanks to its allowance to exchangeinstruments without substantial modifications in software, which allows the user to think about the development of an intelligent distributed measurement system for GPIB equipment.

REFERENCES

[1]. Denis Weznel, Ronald Deichert, Mark Tse, “Emulation and Enhancement of the Honeywell 2600 Automatic Test Station”, ATOTESTCON’99, IEEE-Systems Reading Technology, Conference, 1999, pag. 439- 450.

[2]. P.Daponte, M.Savastano, “Distributed Laboratory Based on Object-Oriented Measurement Systems”, IEEE, 1996, Instrumentation and Measurement technology conference Brussels, Belgium, june 4-6.

[3]. P. Marino, J.Nogueira, H.Hernandez, “Electronics Laboratory Practices Based on Virtual Instrumentation”, ASEE/IEEE Frontiers in education conference, 1999.

[4]. Dany Cheji, “Sofware Architecture for Building Interchangeable Test Systems”, National Instruments, Autotestcon, 2001.

[5]. C.Perello, N.Poch, C.Scroeter, J. Millan, “An Open System to Interface IEEE-488 Measurement Device Designed in a Microelectronics Environment”, Centre Nacional de Microelectronica IMB, IEEE. Freie Universittaet, Berlin 1996.

[6]. D. Grimaldi, L. Nigro, F. Pupo, “Java-BasedDistributed Measurement Systems”, IEEE, Transactions on Instrumentation and Measurement, Vol 47, NO. 1, february 1998.

[7]. J. Odell, “Advanced Object-Oriented analisys & Designing UML”, Cambridge University press sigs books New York, 1998.

[8]. National Instruments, “Measurement and Automation Catalogue”, 1999.

[9]. E. Acosta, F. Baguena, “Descripción Unificada Bom/Ruta/Recursos Mediante el Paradigma Orientado a Objetos”. Instituto Tecnológico de Chihuahua, XXII Congreso Internacional de Ingeniería Electrónica 2000, pag.321-3.