

## VIRTUAL INSTRUMENTS – FUNCTIONAL MODEL, ORGANIZATION AND PROGRAMMING ARCHITECTURE

G.S.Georgiev, G.T.Georgiev, S.L.Stefanova

*Abstract: This paper presents functional model, organization, a programming architecture and an implementation of Virtual Instruments as an essential part of educational laboratory tools. The Virtual Instruments are designed in event- driven programming environment and are capable of performing instrumental functions in local or remote level. The possibility of realization of real time operations from signal information point of view is discussed.*

*Key words: Local Virtual Instrument, Remote Virtual Instrument, Events, Messages, DAQ System, DLL, Sockets, Java RMI*

---

### Introduction

---

Modern measurement systems for data acquisition and processing in engineering and research combine three basic functions:

- DATA ACQUISITION. Usually this comprises a number of measured quantities, characterising the behaviour of the object of measurement; they are sampled simultaneously or sequentially, in most cases multiplexing the measured signals through several analogue channels (8, 16, 32 or more) for further conversion by a common analogue to digital converter. This function is implemented in hardware by DAQ-systems (DATA ACQUISITION SYSTEMS);
- DATA ANALYSIS by means of algorithms for processing the results from multiple, aggregate or combined measurements and specific procedures for measurement and calculation which eliminate systematic errors, depending on the measured quantity and the environment within which it is monitored. Usually this function requires performing a large amount of computational and logical operations for reducing the initial indetermination of the quantity under measurement by comparing it with what is called best value and interval of residual indetermination - standard deviation;
- DATA PRESENTATION. Most often this comprises visual relations among measured data in graphical or table form. Their suitable visualisation has a certain (often decisive) impact on the quality of the carried out measurement process. Usually when conducting engineering or scientific research one has to "experiment" with scaling of graphs, approximation of the processed signals and visualisation of the functional relations.

The implementation of general informational and specific measurement procedures determine the efficiency of the means for carrying out a measurement process and the opportunity for achieving its goals. Virtual instruments [1], as a combination of (quick) hardware and (flexible) software can be part of a well defined teaching hierarchical structure for:

- generation of asynchronous data streams for physical phenomena and properties in the object of measurement;
- classification, discovering of interrelations among them and merging them into a common data base;
- processing of the merged data in order to represent it in accordance with the objectives of the knowledge extraction process and to allow for interaction with the consumer of knowledge.

Building the measuring instruments into a suitable computer environment can carry out efficient accomplishment of these functions. In such a structure, which is new in a qualitative aspect, the computer environment controls the conversion process for efficient performance of analysis and visualisation. So developed computer based instruments can be helpful in advancing of teaching process [1]. In order to promote not only the theoretical but also the practical work of learners in contemporary computer based laboratories there is a trend to integrate the basic theoretical material with simulation, animation, quizzes and practical experiments. Such an approach, in which the experimenting work of learners brings to virtual instruments, is realized in [7]. The applications developed by Asymetrix Toolbook, Microsoft Visual C++ in Microsoft Windows operational environment enable a simulation of measuring procedures. The connections with the laboratory objects – copies of the real circuits, devices and systems the learner is working with in his future professional area are missing. A step forward is the expansion of the standard computer configuration



signals can be equivalent to a common continuous, time invariant system with frequency response  $H_c(\omega)$ , despite the non-time invariants of the pulse modulator [3].

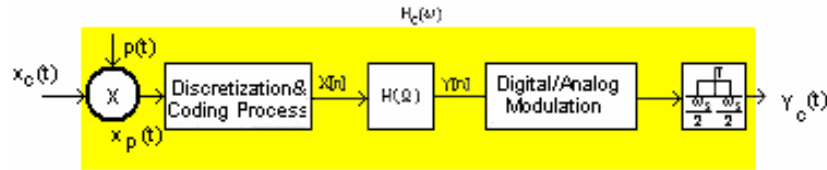


Fig.2 Generic diagram for processing of a continuous signal in a Virtual Instrument

Clarifying the conditions for equivalence is of a crucial importance for correct from an informational point of view carrying out of the measurement process in virtual measurement systems. Modulating the analogue signal  $x_c(t)$  with a pulse train  $p(t)$  with sample period  $T$  and sample rate  $\omega_s$ ,  $T = 2\pi / \omega_s$ , is the most frequently used conversion method:

$$x_p(t) = x_c(t)p(t), \quad p(t) = \sum_{n=-\infty}^{+\infty} \delta(t - nT) \tag{1}$$

The signal  $x_p(t)$  is a pulse sequence with pulse amplitudes, equaling the samples from  $x_c(t)$ , in moments displaced by  $T$  from each other, and frequency equivalent, determined by the convolution of  $X_c(\omega)$  and  $P(\omega)$ :

$$X_p(\omega) = \frac{1}{2\pi} [X_c(\omega) * P(\omega)] = \frac{1}{T} \sum_{k=-\infty}^{+\infty} X_c(\omega - k\omega_s) \tag{2}$$

Consequently,  $X_c(\omega)$  is a periodic function of frequency and accounts for displaced copies of  $X_c(\omega)$ , scaled by  $1/T$ . Restoring the original signal with a limited spectrum  $\omega_M$  is only possible if displaced copies don't overlap, i.e.  $\omega_s \geq 2\omega_M$ . Under this condition, the relation between the frequency spectrum  $X_c(\omega)$  and the signal  $x_c(t)$  is:

$$x_c(t) = \frac{1}{2\pi} \int_{-\omega_M}^{+\omega_M} X_c(\omega) e^{j\omega t} d\omega \tag{3}$$

Decomposing  $X_c(\omega)$  into a Fourier transformation, the following holds true:

$$x_c(t) = \frac{1}{4\pi f_M} \sum_{n=-\infty}^{\infty} x_c\left(\frac{n}{2f_M}\right) \int_{-2\pi f_M}^{2\pi f_M} e^{j\omega(t - \frac{n}{2f_M})} d\omega = \sum_{n=-\infty}^{\infty} x_c\left(\frac{n}{2f_M}\right) \frac{\sin(2\pi f_M t - n\pi)}{2\pi f_M t - n\pi} \tag{4}$$

Equation (4) shows that the analogue signal, defined by samples with frequency  $f_s$ , can be restored by passing the spectrum of the discrete signal through an ideal low-pass filter with cutting frequency  $\omega_c$  and adhering to the more generic limiting conditions:

$$\omega_M < \omega_c < \omega_s - \omega_M, \quad \omega_s > 2\omega_M \tag{5}$$

Failing to observe (5) leads to low-level noise from overlapping of displaced copies' spectrums, also known as *aliasing noise*. This is why incorrect choice of sample frequency leads to unavoidable degradation of the restored signal. Fighting *aliasing noise* should be carried out by precursory limitation of the informative signal's spectrum. For finite in time signals this inevitably leads to the advent of error, which must carefully be considered in preliminary analysis and accounted for in the total balance of errors.

Real-time measurement and preserving of the full information of signals across conversions can be achieved through implementation of local virtual instruments by combining hardware with suitable parameters in the technical part of the measurement system and well considered organization of algorithmical and software resources. In implementing remote virtual instruments, the possibilities for scaling of time are sharply reduced by the time it takes to carry out the routed Internet access, and consequently reduced are the possibilities for carrying out experiments in real-time for most processes under investigation. This is the reason why precursory processing of initially measured data is needed in "immediate proximity" to the object of measurement and integrated forms of estimations should be sent to the client, which, due to generalization, have lost some of the information contained in the measured signals, but have a considerably slower rate of

change. They however can be transmitted with suitable for the Internet medium informational frequencies. It is desirable that remote experiments in remotely accessed labs be described by such integrated ratings, so that the trainee, instead of being a passive observer and recorder can turn to a participant in an interactive monitoring, requiring his or her active share in the experiment, consequently better achieving educational goals.

## Software implementation

One possible architecture of a virtual instrument is presented on Fig.3.

The system is expected to run under Windows, due to the well-documented methods for accessing the hardware in this operating system. Windows imposes the need for a specific module, kernel mode driver, which is the only module that can access hardware directly, e.g. read from / write to IO ports, react to interrupt requests, etc. In this case it incorporates a number of functions, which are used to read data from ADCs, set values for DACs, start timers, etc. The DLL is in fact a wrapper, which makes the functions, implemented by the kernel mode driver available for use by applications. Additionally, as illustrated on Fig. 3, the DLL can also provide:

- a server socket, implementing the server side of a socket based client-server protocol;
- function(s) that can be called via Java RMI;
- Windows-specific synchronization mechanisms for eliminating conflicts if more than one application tries to access the hardware in any one moment.

The former two allow for remote access to the instrument, while the latter allows multiuser access.

This architecture is based on the event-driven paradigm used by Windows and other graphical user interface systems. The system reacts to events by forming messages, which are then placed in affected applications' message queues. These messages are dispatched by a system call, finally reaching a window function - WndProc on Fig. 3. Control is then passed to specific code, processing any message of interest. As mentioned above, events can enter the system as a result of user interaction (from the keyboard, mouse, etc.). The same message-based approach can be used for asynchronous data exchange with the hardware. This allows a remote user to get integrated forms of estimations, discussed in the previous section. If the hardware allows it, the user can request a series of samples to be taken at a given rate. This request is processed in its entirety locally, even strictly at the hardware level. When the required number of samples has been collected, the hardware generates an interrupt request. This is detected by the kernel mode driver, which sends a specific message to the Windows kernel. This message is then dispatched to any application, which cares to process it. Thus the application receives the whole set of samples, taken in real time, and can visualize them locally, or send them to a remote user via RMI or

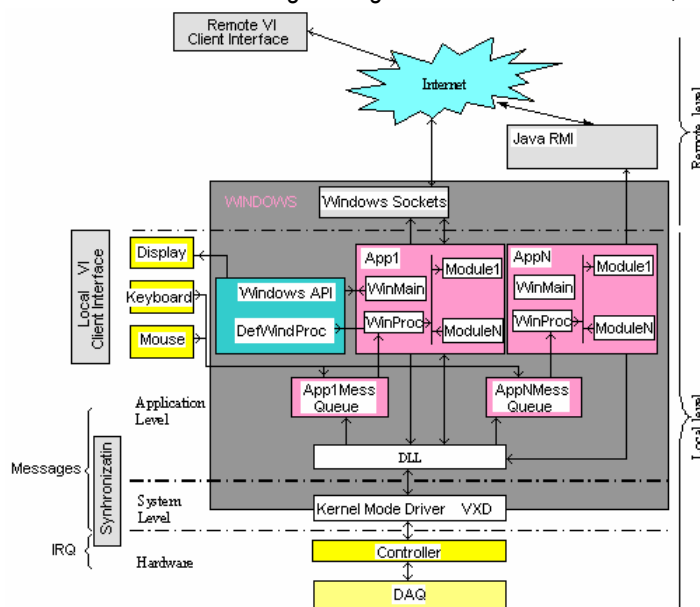


Fig. 3 Software architecture of a virtual instrument

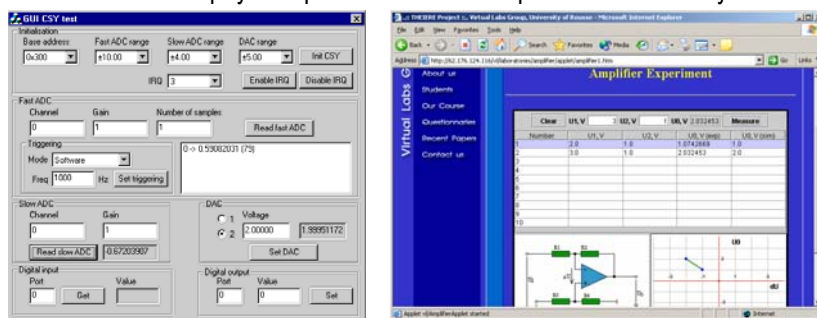
sockets.

Another important feature of this architecture is that all requests for hardware access, whether local or remote, rely on one module, the DLL, for fulfilment of these requests. This makes it very easy to implement synchronization, using standard Windows synchronization objects, such as critical sections or semaphores. Furthermore, if the hardware allows it, the different devices within it (such as DACs, ADCs, etc.) can be virtualised separately and allow concurrent access of more than one user to different devices. The only

requirement to the hardware for achieving such functionality is that it generates IRQ for each individual device and has means to identify which device caused it.

### Realization and conclusion

Using above discussed technology a local test Virtual instrument (fig.4a) for functions control of a DAQ system and a Remote Virtual instrument for analysis of input/output characteristics of Instrumentation amplifiers (fig 4b) are developed. Both of them are implemented as powerful tools in the structure of the Remote Virtual Lab in Rouse University (<http://tie.ru.acad.bg>) [5]. The first is used by teaching staff for verification of the physical part of the virtual experiments - DAQ system KSI 10 [6]. The latter is granted to the



a) b)  
Fig. 4. Virtual instruments as tools of a Remote Virtual Lab

students for a remote training in the field of Electrical Measurement.

This Remote Virtual Laboratory is being constructed in the frame of THEIERE\* Project as the main purpose was the efforts of the "Virtual Lab" group participants to be integrated in order to obtain a didactically structured distributed lab among the

collaborative universities. The authors consider such an approach of teaching process organization as a possibility to make it more intensive and overcoming some constrains of the traditional teaching process concerning time and space limits in practical subjects as Electronics and Electrical Measurement in which practical experiments cannot be replaced by program simulators.

### Bibliography

1. Spiedler H.J.W Virtual Instrumentation and Virtual Environments, IEEE Instrumentation & Measurement Magazine, 1999, pp. 14-19.
2. Spiedler H.J.W, Ulings Virtual Instrumentation: A survey of Standards and their Interrelation, IEEE Technology Conference Ottawa, Canada, May 19-21, 1997, pp. 676-681.
3. Oppenheim A.V., A. Willsky, I. Young Signals and Systems, Prentice- Hall, Inc., 1983
4. Zheliazkova I., G. Georgiev, Organization of Measurement Systems in an Event Programming Environment, Automatica and Informatics Journal, N6,1999, pp.44-49.
5. Georgiev G., E.Stoyanov, S. Stefanova, D.Stoykov, H.Roth, Step by step towards the Virtual Lab organization, Int. Journal "Information Theories&Applications", 2002, vol.9, pp.153-159
6. DAQ system KSI 10, BSA, ITCR, Sofia, 1988
7. Pisani U., F.Cambiotti, N.Sala, F.Sanpiero, A Hypermedia solution for electronic instrumentation measurements practice, Proceedings of the 3<sup>rd</sup> International Conference on Computer Aided Engineering Education, 13-15 September, 1995, Bratislava, Slovaki, pp.144-149.
8. Majumdar S., C.Mazidar, D.Ray, On line CAI on Microprocessor Interfacing, Proceedings of the 3-rd International Conference on Computer Aided Engineering Eduvation, 1995, Bratislava, Slovakia,pp.144-149.
9. Ponta D., G.Donzellini and G.Parodi, "Practical electronics taught by hypertext: the WORKBENCH project", Int.J.Computers & Education 16(1), 1998, 127-132.
10. Ponta D., G.Donzellini, "Learning electronics with hypermedia and computer tools", Proceedings of the International Conference on Computer Aided Learning and Instruction in Science and Engineering, August 31, September 1-2,1997, Paris, France, pp.7-23.

### Author information

G.S.Georgiev, G.T.Georgiev, S.L.Stefanova - Rouse University, 7017 Rouse, 8 Studentska Str. e-mail: [gsg@ru.acad.bg](mailto:gsg@ru.acad.bg)

\* Thematic Harmonisation in Electrical and Information EngineerRing in Europe,Project Nr. 10063-CP-1-2000-1-PT-ERASMUS-ETNE