Computer Interface for Learning and Using Virtual Instrument

Yongkai Fan[1], Tianze sun[1], Jun Lin[2], Xiaolong Fu[1], and Yangyi Sui[2]

[1] Computer and Infromation Management Center, Tsinghua Uinversity. 100084, Haidian District, Beijing China
{fyk,stz,fxl}@cic.tsinghua.edu.cn
[2] College of Instrument and Electrical Engineering, Jilin Uinversity. 130026, Changchun, Jilin Province, China
{lin_jun,syy}@jlu.edu.cn

Abstract. Computer interfaces representation, design and implementation as the computer software outward window had a large impact on software learning and using, especially for virtual instrument. Several types of virtual instrument developing software are available for the virtual instrument development and parts of them have great influence in the instrument science field. However, only a limited number of inexperienced or previous untrained people are able to well utilize them. Part of the limitation stems from the difficulty in learning how to use them and part of from the demand of software developing expertise background or hardware design abilities. Therefore, user friendliness of virtual instrument software is needed for a great number of people who are without expertise background or hardware design abilities. There are a number of features in the software described in this article would server to meet the need, such as requirement-driven idea from human-computer interface implementation aspect helped step obstacles encountered by users with limited experience. An experimental interface design has been developed to use an advanced object-oriented development environment, thus allowing a great deal of flexibility in implementing changes and adding new features in order to provide a friendliness operation interface to actual users.

Keywords: Human-computer interface, Virtual instrument, Module instrument.

1 Introduction

Widespread application of personal computer and advent of new technologies in computer science over the past two decades has given rise to a new world in present measurement and instruments filed--virtual instrument. The importance of virtual instrument in measurement and instrument filed results in it as a compulsory course appeared in engineering students' curriculum and as a substitute for traditional instrument [2, 4, 7]. Virtual instrument programming environment, tool, and even graphic languages rush to support the process of learning virtual instrument. The graphical programming software package LabVIEW (laboratory virtual instrument engineering workbench, example of one rare panel of LabVIEW is shown as fig. 1)

C. Stephanidis (Ed.): Universal Access in HCI, Part III, HCII 2007, LNCS 4556, pp. 569–577, 2007.

has made the concept of the virtual instrument (VI) a practical reality [1, 3, 5, 6]. LabVIEW gradually becomes the substitute name of virtual instrument. However, it is a difficult subject for beginner to learn and program a virtual instrument, especially one without any related software programming experience.

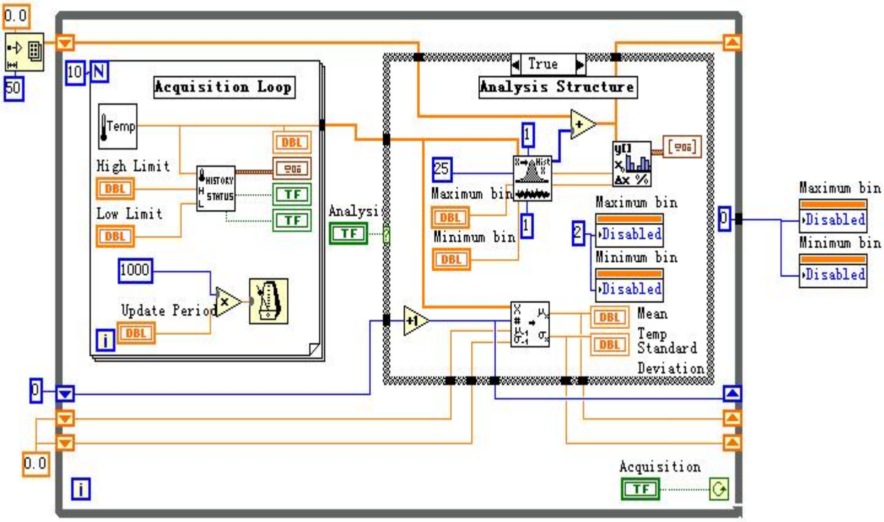

Fig. 1. One example of rare panel of LabVIEW

We are actively exploring ways to use current technology, including computers and the friendly human-computer interface, to aid in instruction and using virtual instrument in an easy way. To this end we are developing a virtual instrument developing system where people can be exposed to easy way to understanding virtual instrument and methods of data collection and processing which they otherwise might not clear because of limited experience and not the expert in software/hardware design in virtual instrument. Herein we put forward a new way to study and use virtual instrument via a system called module instrument(MI), people can use the system to learn how to set-up and operate the equipment and work with the actual device. This system relieves people from the traditional way by which people who intent to a virtual instrument must study the principle of how to design it and understand the working theory of this type of virtual instrument. The system's easy-to-use interface are hence designed and run experiments for people using virtual instrument without experiences rather than for training the users as professional software designers.

1.1 Concept of Virtual Instrument

The concept of virtual instrument is put forward by national instrument corporation: "a virtual instrument is a layer of software and/or hardware added to a general-purpose computer in such a fashion that users can interact with the computer as though it were a traditional electronic instrument." The computer screen presents a rendition of the control panel of the instrument with controls. The user manipulates the controls with a mouse. The instrument displays readings from signals acquired through a hardware interface to the external world. Commonly, the accepted scheme of virtual instrument is shown as fig. 2

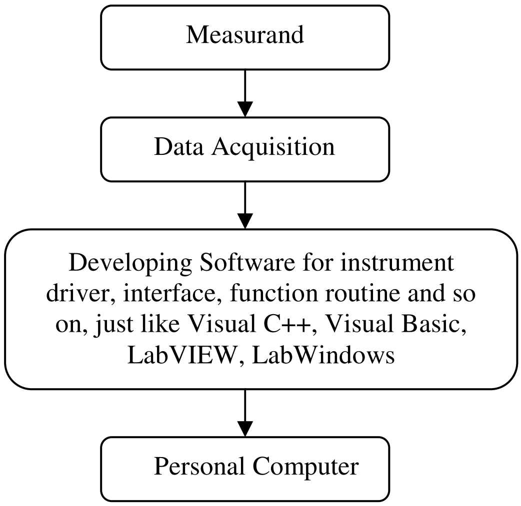

Fig. 2. Scheme of Virtual instrument

1.2 New Way to Virtual Instrument

Instead of spending a lot of time to understand the programming grammar and the development environment just learning how to use the software to control and operate the instrument and to study the working theory of virtual instrument, any individual who just wants to find a simple way to use simple virtual instrument's function, such as data acquisition, may utilize this new way to develop virtual instrument without hardware or software experience needed in other virtual instrument developing environment. With this new tool, people do not acquire necessary experience to debug and program virtual instrument since they can be quickly adept at handling the software to control the actual instrument.

To this end, one new virtual instrument developing environment MI was put forward.

2 Module Instrument

MI is a virtual instrument developing environment (tool), it is devised a module-based framework, designed to allow user who without any previous experience and training to generate virtual instrument easily and naturally, to learn the basic principle of virtual instrument. In order to provide easy understandable interfaces for inexperienced people rather than specialist, MI employs function tightly encapsulated.

The primary consideration of MI is how to encapsulate some common used functions as modules for construct VI, and then how to realize VI via only few required module selection.

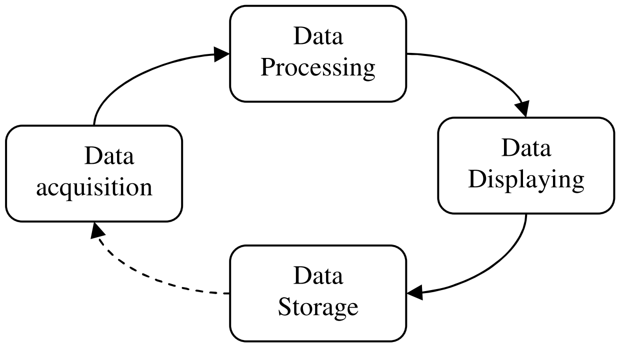

Fig. 3. The common data flow sequence between different modules: the dotted line means that the data flow cycle may be one time or more than one

To encapsulate function as modules, all the function encapsulated can be classified into four types of module: acquisition, displaying, processing and storage. The classification of modules is based on the common data flow sequence when VI works (show as fig.3). All those modules are shown as icons on the toolbar under the environment of module instrument. When user wants to realize one type of VI, he/she just selects corresponding module form those four types modules, and the MI system can automatically connect the selected modules to generate the virtual instrument.

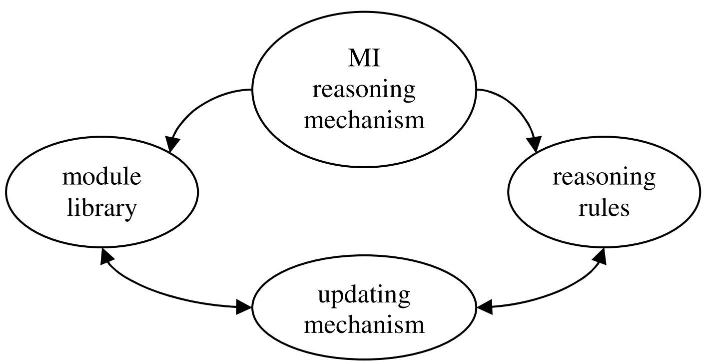

Fig. 4. Relationship scheme of four components for MI system

For the purpose of generate VI based on the module selection, MI system consists four main components: MI reasoning mechanism, module library, reasoning rules and updating mechanism. The relation among those four components is shown as fig. 4: 1)

MI reasoning mechanism is the most important part for MI, this component joints selecting modules to generate VI according to the reasoning rules and is responsible for the coordinating among different modules. 2) Module library is the collection of four types of modules mentioned before and presented as toolbar in MI. 3) Reasoning rules are the basement for MI reasoning mechanism, MI reasoning mechanism uses those rules to determine whether using the selected modules to generate VI is feasible or not, and if feasible, determines the data flow sequence among selected modules.

3 MI and Requirement-Driven

The main aim of MI is simple and easy to use, so how to expose the MI modules to user in an obvious way should be taken into account, requirement-driven [9] is just the idea to tackle this question. Requirement-driven refers the generation of VI is based on the user's most directly requirement which is the foundation of VI generation and is the information source from the user. The main point of requirement-driven for VI generation is relieving user both form complex programming grammar studying and from the deeply understanding instrument inner structure, and the only thing users have to do is putting forward the functional requirement. The idea of requirement-driven for MI is described in fig. 5.

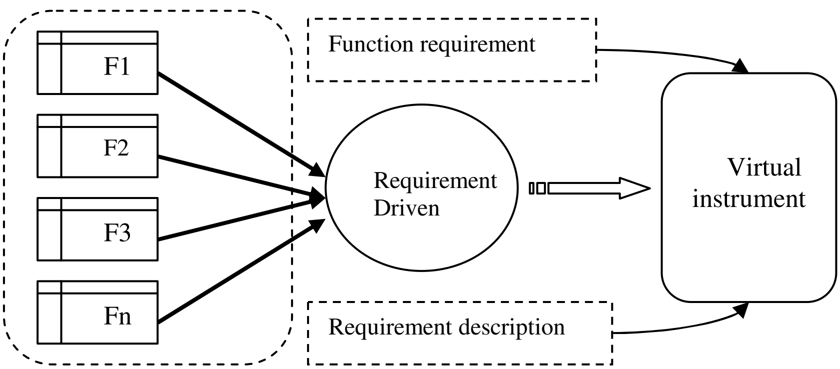

Fig. 5. The exploded map for requirement-driven. Letter F represents function description.

From inexperience users' viewpoint, the simpler one is better. To realize the requirement-driven in MI, the fulfillment of requirement description is just the several times mouse clicks. Based on the highly encapsulated modules, four types of modules are designed as four toolbar user interface in MI and each module is shown as icon on toolbar with the help tip when mouse moves on it. By this way, the use can select one module from those four toolbar respectively; the MI will generate a VI or give selection errors according to the default reasoning rules. Fig.6 shows a data generator from acquisition toolbar in which all available acquisition modules lie.

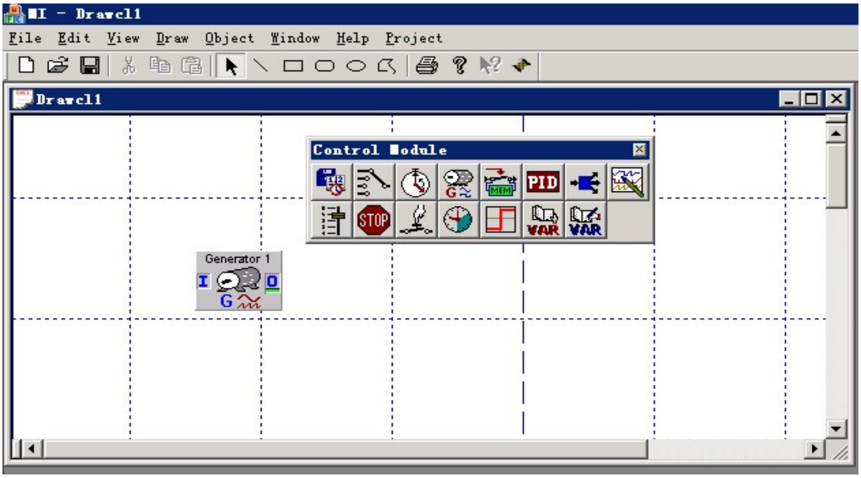

Fig. 6. The selection operation interface for user

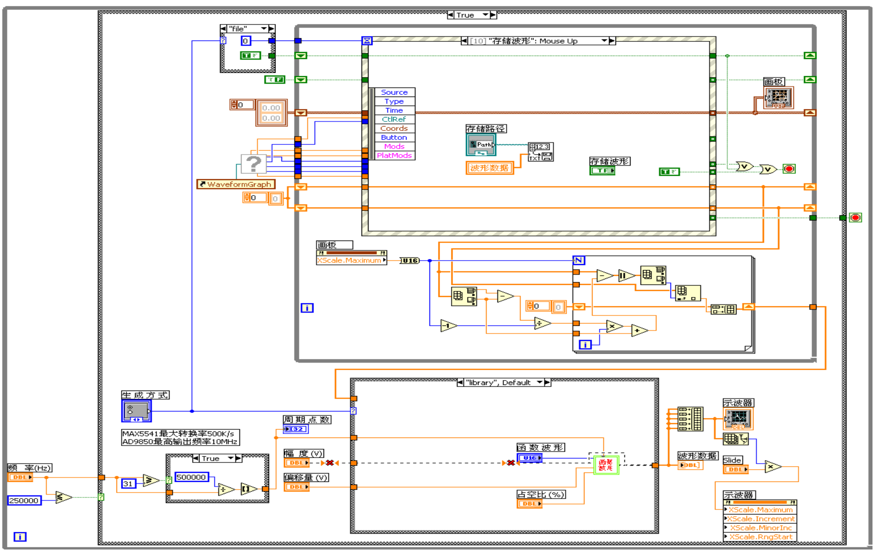

Fig. 7. Screen shot of rare panel for virtual digital oscilloscope under LabVIEW environment, the line connect different type of icon is the idea of data flow. Different type graphic means different functional units.

4 Example of Virtual Digital Oscilloscope

In this section, we present two examples with the same function: virtual digital oscilloscope which displays the simulated waveform. One example of virtual digital oscilloscope is designed by LabVIEW, and the other by MI. Through those two

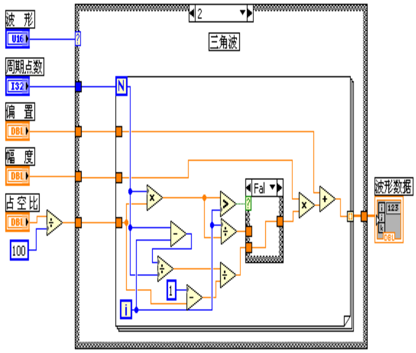

Fig. 8. The rare panel screenshot of design for triangular wave

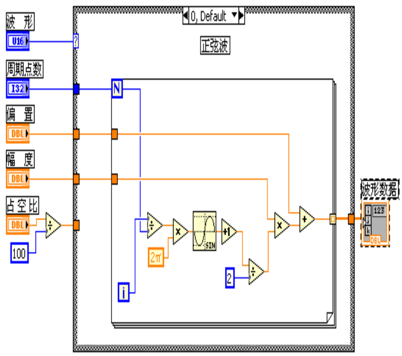

Fig. 9. The rare panel screenshot of design for sine wave

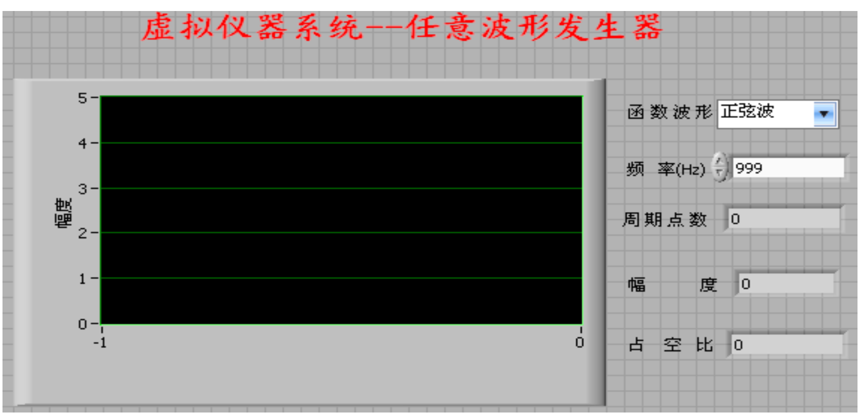

Fig. 10. The front panel screenshot, which is also the user interface of VI

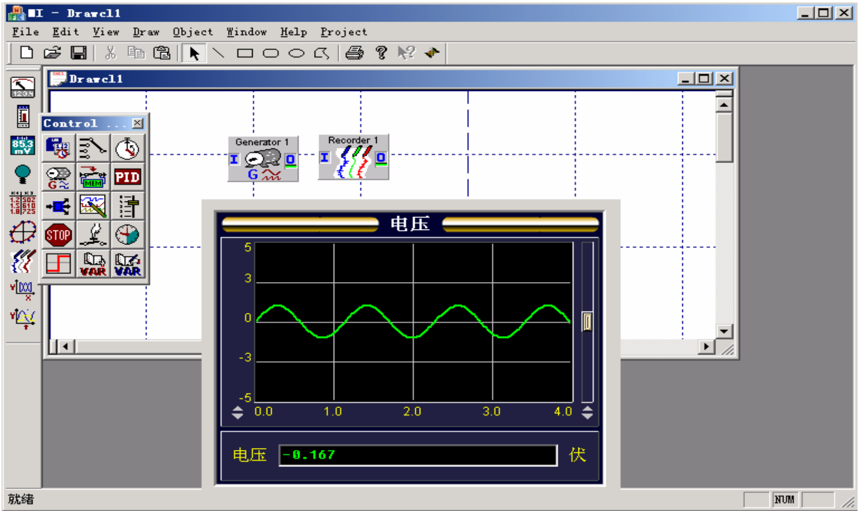

Fig. 11. The screenshot of virtual digital oscilloscope designed by MI

examples, we can get the general impression of MI's objective and user- group faced. The comparison between MI and LabVIEW means to explain why laypeople can use MI easily realizes common functional VI. Fig.7 is the rare design panel of virtual digital oscilloscope under LabVIEW environment; Fig.8 and 9 are the detail design of triangular wave and sine wave; Fig. 10 is the interface of this VI under LabVIEW. Fig.11 is the interface and VI generation under MI environment. From the comparison, MI provides a more easy-to-use interface to realize virtual digital oscilloscope than LabVIEW and gives inexperienced people a more simple way to realize VI and know the VI's working flow from the two type of module selection (one module is the data generation module from acquisition toolbar and the other is the data display module from displaying toolbar). The generation of virtual digital oscilloscope only needs two modules and just two time mouse clicks.

5 Conclusion

In this paper, we introduced a virtual instrument developing environment called MI which exploits a novel reach to virtual instrument using and learning for inexperienced people in related field. The long series of experiment test and evaluation reports from inexperienced people and people who are familiar with LabVIEW encourage us to access that MI is possible to transfer to popularity among untrained people. Unfortunately, the generation of MI is limited. The VI from MI can be the replacement of common used traditional instrument and can not be used to realize very complex functional virtual instrument. Also, there has been no testing of the usability of cross-platform; MI is just can be run on windows operation system. Much more study of this area is sorely needed. In addition, there needs to be a study of how MI reasoning mechanism can apply to generation VI based on many modules selection. Current work has been limited to a few modules, MI reasoning mechanism can work well if the selected modules are no more than two in each type. In order to make MI simpler, much more technique details about human and computer interaction must be considered under MI. Future work will involve the develop of a prototype with a more intelligent user interface.

References

1. Wong, W.: LabVIEW embraces graphical object-oriented programming. Electronic Design, vol. 54(21), pp. 38–40. Penton Publishing Co, Cleveland, OH 44114-1503, United States (2006)
2. Leping, Y., Haitao, L.: The application of virtual instrumentation in aerospace engineering education, Shanghai (2001)
3. Anon. LabVIEW 20th Anniversary edition emphasizes communications test. Electronic Design 54(18), 33–34 (2006)
4. fan, Y, Lin, J.: A Wireless LAN-Based Robust and Scalable Virtual Laboratory for E-learning. In: Pan, Z., Aylett, R., Diener, H., Jin, X., Göbel, S., Li, L. (eds.) Technologies for E-Learning and Digital Entertainment. LNCS, vol. 3942, pp. 322–325. Springer, Heidelberg (2006)
5. Elliott, C., Vijayakumar, V., et al.: National Instruments LabVIEW: A Programming Environment for Laboratory Automation and Measurement. JALA - Journal of the Association for Laboratory Automation 12(1), 17–24 (2007)
6. Navaee, S.: Computing and programming with LabVIEW, Salt Lake City, UT, United States, American Society for Engineering, Washington, DC 20036, United States (2004)
7. See, A.: Hands-on learning and implementing using LabVIEW for undergraduates in 13 weeks, Salt Lake City, UT, United States, American Society for Engineering, Washington, DC 20036, United States (2004)
8. Sevgi, L., Uluisik, C.: A labview-based virtual instrument for engineering education: A numerical fourier transform tool. Turkish Journal of Electrical Engineering and Computer Sciences 14(1), 129–152 (2006)
9. Yong-kai, F., Jun, L., Tian-ze, S.: Program generation mechanism based on module relationship match reasoning. Journal of Jilin University(engineering and technology edition) 36(6), 939–944 (2006)