A&QT-R 2002 (THETA 13)
International Conference on Automation, Quality and Testing, Robotics
May 23 – 25, 2002, Cluj – Napoca, Romania

ACTIVEX TECHNOLOGY IN VIRTUAL INSTRUMENTATION

Tarța Adriana - Mihaela

S.C. IPA S.A. Cluj, e- mail : adriana@automation.ro

Abstract: this paper try to discuss the possibility and the methods for making the Internet active for virtual instrumentation by means of ActiveX controls.

Key words: control, Internet, share, time

1. INTRODUCTION

It's just not possible to give a clear technical definition of what the term ActiveX means. The reason for this is simple: ActiveX is a marketing label, not a technical term. There are many technologies grouped under this label, and exactly what those technologies are changes over time. Still, all of them have something in common--they all use COM.

ActiveX controls are among the many types of components that use COM technologies to provide interoperability with other types of COM components and services. ActiveX controls are the third version of OLE controls (OCX), providing a number of enhancements specifically designed to facilitate distribution of components over high-latency networks and to provide integration of controls into Web browsers. These enhancements include features such as incremental rendering and code signing, to allow users to identify the authors of controls before allowing them to execute.

When the term was first introduced in early 1996, it was used to refer to technologies that were somehow associated with the Internet and the World Wide Web, and the label still retains much of that flavor. Today, though, Microsoft includes COM itself as part of ActiveX, and there's nothing at all Internet-centric about COM.

ActiveX controls, formerly known as OLE controls, are software components callable from many different development tools, including Visual C++, Visual Basic and Access. You can think of an ActiveX control as a self-contained executable code module for performing a task that can be controlled from a programming environment.

Today, ActiveX controls are available for acquiring data from plug-in DAQ boards, analyzing data and building virtual instrument panels.

2. ACTIVEX IN VIRTUAL INSTRUMENTATION

Many scientists and engineers thought that the Internet has nothing to do with automated test systems involving data acquisition (DAQ) and instrument control, but they were wrong.

As the test industry becomes increasingly reliant on the personal computer (PC) for automated solutions, any new technology affecting the PC, like the Internet and World Wide Web, will eventually work its way into the virtual instrumentation world. Many test engineers already recognize the Internet as a key support tool for development teams. Advances in Internet technology also are making common instrumentation tasks, such as remote monitoring and control, trivial.

Intranets are private networks inside a company or organization that use the same kind of software you would find on the public Internet, but is for internal use only. It uses the same easy-to-use tools, such as web browsers and HyperText Markup Language (HTML) files with dynamic links and jumps, that are available for the World Wide Web.
An Intranet server makes an ideal document-distribution system. Many test-engineering teams distribute source-code files, test methods, fixture instructions and user manuals from an Intranet server. A web browser can easily access the files from any computer platform or operating system. System administrators also can track file use through standard web-reporting methods. With the flexibility of today's server tools, an Intranet server can act simply as an easy-to-use graphical front end to a standard source code control and tracking repository.

Like the Intranet, the Internet and its easy-to-use access tools make a good platform for distributed and remote virtual instrumentation. With the Internet, you can distribute completely running applets to field-service personnel or pass data from a remote data-monitoring system back to the corporate offices. More importantly, these capabilities are available through easy-to-use development tools. Through ActiveX controls and new web-page development tools, Microsoft makes remote virtual instrumentation a reality.

How are ActiveX controls related to the World Wide Web? The latest group of tools to support ActiveX controls is web browsers. Both Microsoft Internet Explorer and Netscape Navigator (through an add-in) can view HTML sheets with embedded ActiveX controls.

As a result, you can download an HTML page and run it within your web browser. This means you can build test programs, or applets, directly into HTML sheets with a delivery system as simple as a web browser.

You might have an ActiveX control for acquiring data with a plug-in DAQ board, another for displaying data on a graph control, and a third for sending data over a transmission control protocol (TCP) to other computers. The VB-Script might trigger the DAQ control to acquire a waveform from a particular input channel and then send the data to the graph control. Because the applet executes locally within a browser, end users running the program can operate the user interface controls as a dynamic front end.

For example, once the data is passed to the graph control, you can zoom in on the data, operate cursors and pan on the data in the graph. Likewise, knobs, meters, gauges and push buttons can be operated directly from within the browser.

Once you acquire the waveform and analyze it in the graph control, press a button to send the data back to a centralized file transfer protocol (FTP) server using the TCP control.It is significant that you can write a program that acquires a waveform, plots it on a graph, and then sends it back to a remote computer for the delivery mechanism and the distribution requirements You only need a web browser (Internet Explorer or Navigator with an add-in) to run the program as well as a DAQ board installed on your computer.

. The program is an HTML page that is downloaded just like any other web page. This is an ideal scenario for field-service tests. Rather than equip your field-service personnel with a large collection of programs that will need to be maintained on each of their hard drives, simply provide the maintenance programs on a web server.

Whenever field technicians need a program to diagnose or to service a unit in the field, they connect to the web server and download the appropriate one. As the test-program developer, you maintain one set of files on the server.

3. FUTURE DEVELOPMENTS

Today, most of the ActiveX web-based technology focuses on the client, meaning clients can download web sheets and execute them on the local computer. In the future, Microsoft will apply the VB-Script concept for controlling server-side operations. This means you can develop more advanced servers that can query clients connecting for particular information and make decisions on the fly based on the client input. These decisions will be driven by VB-Script.

As general-purpose PC technology continues to evolve and develop, astute scientists and engineers will be able to find faster, easier and cheaper solutions to traditional test and measurement challenges. The Internet and World Wide Web are just examples in which technology is easily adapted to meet the needs of the engineer—in this case, for remote virtual instruments. In the case of the Internet, the key to its success in the virtual instrumentation arena is ease of use—of the end-user tools such as web browsers in general, and of the new development tools such as the ActiveX ControlPad.

Microsoft is not the only software company taking advantage of the web. In fact, many of the leading instrumentation software vendors now offer web connectivity over the Internet.

For example, with the new Internet Developers Toolkit, you can add web-server capabilities to your data-acquisition and instrument-control applications built in LabVIEW or LabWindows/CVI. Following this server scenario, multiple clients can connect to the LabVIEW application to download a replica of the application's front panel.

The web server automatically supplies images of the virtual instrument front panel as a single snapshot or as an animation. Through the Internet Developers Toolkit, you can make any virtual instruments immediately accessible over the Internet from any other computer equipped with a web browser and Internet connection; no extra programming is required.

4. COMPONENTWORKS

National Instruments has realized one of the most popular development environments for virtual instrumentation, LABVIEW. This has the advantage that doesn’t need programming knowledge, being dedicated in principle to engineers and scientists.

People who has experience in programming and want to use the instruments available by programs like LabVIEW, can use LabWindows, an programming environment compatible ANSI C, dedicated to C programmers. If you are Visual Basic, Visual C++, Delphi programmer or you want to build an internet page for data acquisition or to view the parameters of a process that is developing miles away, National Instruments offers

ComponentWorks, a collection of ActiveX controls, compatible with Visual Basic, Visual C++, Delphi, Internet Explorer or any other programming environment compatible ActiveX.

ComponentWorks is a collection of virtual instrumentation add-on controls and libraries that extends Visual Basic from a general-purpose programming tool to a virtual instrumentation development environment. ComponentWorks consists of four major functional components for building virtual instruments: data acquisition controls; GPIB instrument drivers; analysis libraries and user interface controls.

ComponentWorks is a collection of 32 bit ActiveX controls designed for scientists building virtual instrumentation systems. With ComponentWorks, you combine the power and flexibility of standard development tools, such as Microsoft Visual Basic and Visual C++, with the instrumentation expertise of National Instruments. Based on ActiveX technology, ComponentWorks controls are easy to configure through simple property sheets and easy to control from your programs using high-level properties and methods.

Using ComponentWorks, you can develop complex user interfaces for data visualization, for controlling acquisition board From National Instruments and analyze data received or acquired from other sources. ComponentWorks package contains the following instruments:

- Controls for user interface – ActiveX controls on 32 bits for data presentation in a technical format. These controls include: knobs, graphs, cursors, sliders, tanks, thermometers, recipients, buttons, dials, LEDs and binary switches.
- DAQ Controls – ActiveX Controls on 32 bits for data control and acquisition from analogical input/output, digital input/output, counters and temporizer of acquisition boards From National Instruments.
- GPIB Controls, Serials and VISA – ActiveX Controls on 32 bits, for data control and receive from the instruments connected to GPIB interfaces, serials or VXI of computer. GPIB instruments must be connected by the mean of a GPIB interface from National Instruments.
- DataSocket Controls – ActiveX Controls for data sharing and change between applications and a number of other destination, including applications, files and FTP or Web servers.
- Controls for analyze – statistical functions, signal processing, filter, algebra routine, matrix and probabilities.
- Drivers for instruments (only in complete form) – source code and DLL libraries on 32 bits for GPIB instruments controls.

ActiveX controls are made for Visual Basic work. Some facilities of ComponentWork were made considering specific things of Visual Basic. Many interface and analyze controls are based on the similar routines from LabVIEW and LabWindows-CVI, two development environments well known in the world of instrumentation programs developers.

4.1. User Interface Controls

These controls are made for realize special interfaces, specific to instrumentation. Technical applications have specific requirements for user interfaces that cannot be satisfied by similar controls from Visual Basic, Visual C++ or Delphi.

ComponentWorks controls are made for satisfying performance and functionality requirements of technical application, like: displaying waveform with big speed, auto-scaling axes, logarithmic axes, engineering measurement units.
ComponentWorks has a wide variety of controls that helps us building applications.

You can display in real time waveform acquired from DAQ or other instruments, or you can view data that changes very slowly, like temperature, and pressure. The display controls can display many signals in the same time, column graphics, zoom, multiple Y axes and interactive cursors.

As a response to user action, like pressing a button or moving cursor in a graphic, the control generates an event to the application that is running. You only have to add the code that treats that event. Every control is configured using a properties sheet.

You can set the apparition, operation mode, labels, the domain and the format of controls that are used. These properties can be modified in the running time, if it is necessary. The image displayed by controls can be modified by importing images in bitmap format. One defined, a control can be saved into a file and used by another application.

The functionality of ComponentWorks controls is based on events. In standard acquisition operations the program execute all the tasks, including data reading from input buffer and output buffer update. Using methods based on events, acquisition operation are configured and started by the program, but the hardware and software driver NI-DAQ announce the application about new data arrival or providing new data for output operation. Responding to this event, the application executes the code corresponding to the event.

The program don't have to monitor de acquisition process, so it dispose of lot of time for other processes, so it is more efficient and the process of event treatment is invisible for the user. Analyze routines convert the acquired data into results that can be interpreted like conclusions that can improve the quality of tested products. With the aid of the base package you can make statistical analyzes, operations with vectors and matrix and also operations with complex numbers.

For networks, ComponentWorks offers DataSocket controls, dedicated to access, share and save data. Using this control you can load data from files found on Web or FTP servers, or change data with other application anywhere in the world.

You can also create Web sites, so you can display and analyze the data acquired. Also, there are made available, as separate packages, controls for image acquisition and symbols processing for automation, PID controls and PLC drivers. No matter the test or automation system that you want to build, ComponentWorks represents a solution for fast and simple competitive applications development.

5. MEASUREMENT STUDIO

With National Instruments Measurement Studio, engineers and scientists gain the Microsoft Visual Basic and Visual C++, and ANSI C package LabWindows/CVI measurement tools they need to develop test and measurement applications in the programming environment of their choice. Measurement Studio users gain data acquisition, analysis and presentation functions to create measurement applications for stand-alone

measurement systems or distributed measurement over Web. With LabWindows/CVI and ComponentWorks packages users gained measurement capabilities for ANSI C and Visual Basic languages. With the Measurement Studio package, you have the flexibility to choose which industry standard environment best suits your application.

Measurement Studio takes advantage of Microsoft ActiveX/COM technology, the standard for component technology and code interchangeability and integration. Component Object Model (COM), the standard for client/server communication in Windows creates binary compatibility between different code from different compilers by specifying a standard interface that all code modules must support. ActiveX takes COM further, delivering an easy-to-use property page interface for configuration at design time, a user interface to the object and flexible events and methods for programmatic control. You can use Measurement Studio ActiveX controls for data acquisition, instrumentation, analysis and presentation within any control container, such as Internet explorer or Excel. National Instruments developed Measurement Studio ActiveX controls with the Visual Basic in mind.

Measurement Studio offers environment specific tools for easy data acquisition, so that you don't have to do low level programming to set up your acquisition routines or transfer buffers from your computer to your data acquisition boards. Some functionality offered are : single point analog input – acquire a single point analog, waveform analog input – acquire continuous analog buffered waveforms, single-point digital input/output – input/output single point digital updates, pulse generation – generate single, finite and continuous pulses, buffered and double buffered data acquisition and waveform generation – sample and process data or generate waveforms at maximum sampling rates, SCXI – control channel scanning with SCXI modules and convert voltages measured from thermocouples, RTDs, thermistors, and strain gauges to appropriate units, Self-calibration – with self-calibration circuitry, adjust the gain and offset for the board within specified accuracy so you do not have to adjust trimpots to calibrate the board.

6. CONCLUSIONS

Using ActiveX, Component Object Model (COM), and Dynamic Link Libraries (DLL), engineers can integrate applications to extend the capabilities of their measurement system or share applications across different languages to save valuable development time and prevent costly duplication. LabWindows/CVI users now can incorporate ActiveX controls from other software programs or add capabilities, such as streaming video or voice control, to their Measurement Studio application without having to develop technology in-house to support the new feature.

6. REFERENCES

[1] http://www.adtest.co.za/NI%20Software%20Component%20Works.htm
[2] http://www.chappellassoc.com/articles/article_Intro_ActiveX.html
[3] http://www.chappellassoc.com/articles/article_ActiveX_v_JavaBean.html
[4] http://www.netreport.ro/pcrep83/031.shtml
[5] http://www.evaluationengineering.com/pctest/articles/e702pcni.htm