Increase Productivity with an Integrated Software Framework for Measurement and Automation
Introduction

In an “ideal” world, a company would not need to test a product as it evolves from research to design and through production. However, the fact is that measurement systems are an integral part of the product development process. Companies require, and even demand, measurement systems that are a strategic asset in meeting goals for improved quality, faster time to market, greater engineering and manufacturing efficiency, and, of course, lower costs.

In the past 20 years, to meet customer demands, measurement systems have gone through a fundamental change – a Measurement Revolution. This revolution has been driven by a new system architecture in which the computer is at the heart of measurement systems.

The Measurement Revolution transformed test, measurement, and automation applications from loosely coupled, and often incompatible, stand-alone instruments and devices to tightly integrated, high-performance measurement and automation systems. At the core of this revolution lies a component that has become increasingly more important – software.

While the hardware advances in the personal computer have driven significant performance improvements and cost reductions in measurement systems compared to traditional stand-alone instruments, it is the highly productive, integrated software that empowers hundreds of thousands of engineers and scientists to take advantage of these benefits. A complete set of software tools – from measurement driver architectures and instrument connectivity software to highly productive application development environments and open connectivity with standard tools across the enterprise – gives engineers and scientists the freedom to create a new level of powerful, customized measurement systems.

Software has driven the adoption of high-performance, low-cost, tightly integrated measurement systems throughout all areas of the product design life cycle – from research and design, to validation and verification, to manufacturing and test, to service. Worldwide, companies gain a competitive edge by using computer-based measurement and automation systems to design and test higher quality products at lower costs and in less time.
More than two decades ago, as communications buses such as RS-232 serial and GPIB (IEEE 488) were introduced, the computer first became an integral part of measurement systems. By connecting the measurement device directly to the computer (see Figure 1), engineers and scientists could reduce the time-intensive, error-prone process of manually transferring data to a computer for further analysis. In addition, by using the computer as a central “controller” for all instruments in a measurement system, they could coordinate or automate several instruments into a single system.

A software interface to send commands to and receive responses from the instrument brought widespread acceptance of this computer-centric instrumentation system. This instrument control software delivered a driver that installed as part of the operating system, as well as a standard API for consistency across platforms and programming environments – a critical feature considering the significant changes in computing platforms in the past 20 years.

While computer-based instrument control brought benefits to engineers and was a fundamental part of “rack-and-stack” systems, the system development was still painstaking due to the need for detailed knowledge of the command syntax for every instrument or device in the system. Developers tried to standardize command syntax, but what really took development productivity to the next level was the introduction of instrument drivers as part of a new generation of application development environments, such as National Instruments LabVIEW™ graphical development environment, and LabWindows™, a tool that incorporated measurement-specific functionality with C and Basic programming languages. Today, with thousands of instrument drivers available, engineers can focus on high-level measurement system needs rather than low-level, vendor-specific command syntax.
By the mid-1980s, vast improvements in bus architectures led to the development of modular measurement devices that users could install in the computer (see Figure 2). Rather than relying on expensive dedicated processors, firmware, and memory inside a stand-alone instrument, the plug-in data acquisition board took advantage directly of the built-in computer components to deliver smaller, more cost-effective, and higher performance measurement systems.

However, this low-cost, flexible platform required more from the system software than just sending commands and receiving responses. High-level application programming interfaces (APIs) simplified the process of rapidly streaming data from the data acquisition board into computer memory. With high-speed signal processing algorithms and software tools, engineers and scientists created their own custom analysis routines. Design software for user interfaces brought the graphs, knobs, and sliders of traditional stand-alone instruments to the computer screen. Measurement-specific development environments brought unparalleled productivity gains to engineers – delivering tightly integrated tools for instrument control and data acquisition, data analysis, and data visualization.
In the past five years, bus architecture innovations such as PXI™/CompactPCI™ created a preferred platform for highly sophisticated measurement and control capabilities. Unlike traditional instrumentation systems, today’s integrated measurement and control systems consist of a wider variety of measurement devices. This process started with connectivity to traditional stand-alone instruments, then added plug-in analog and digital measurement devices, and today includes image acquisition devices for visual inspection and motion controllers for machine control (see Figure 3). Additionally, these integrated measurement and control systems must easily deliver connectivity to numerous devices such as programmable logic controllers (PLCs).
**Today’s Challenges**

Today, engineers and scientists still face many challenges as they integrate automated measurement systems. With the constant pressure to bring new, higher quality products to market faster, system developers face the challenge of quickly building automated measurement systems to validate designs and test finished products, deploying those systems to numerous locations with minimal downtime of existing processes, maintaining those systems over time – even with changes in staff – and modifying those systems quickly as new designs are introduced. Recent trends to integrate these measurement systems into enterprise systems only add to the complexity that developers face.

**Integration of Diverse Measurement Devices**

Today’s stand-alone instruments, like the traditional instruments 20 years ago, are optimized for interactive, manual use rather than for tightly integrated systems. Because these stand-alone instruments are optimized for interactive use (the manual turn of a knob on the instrument’s front panel or the manual selection between measurement modes), integrating them into an automated measurement application often requires sending the instrument numerous software commands. Not only does this degrade total system performance, but also it results in a loss of development productivity.

In addition, integrating numerous instruments into a single system is often time-consuming and difficult. Synchronizing multiple measurements from different instruments often requires extensive software programming. Measurement devices with different communications mechanisms (such as GPIB, USB, and VXI) require system developers to learn numerous APIs and design programs differently for each type of instrument. These challenges result in lost time for developers who must integrate instruments and understand communication idiosyncrasies rather than focus on solving the measurement problems that prevent delivering their product to market faster.

**System Deployment, Maintenance, and Modification**

Today’s measurement system developers require better tools for system deployment, maintenance, and modification. Because developers now measure product design modifications in months rather than years, they must reduce the time required to deploy a test system to numerous locations. Furthermore, it is no longer possible to write new systems from scratch each time a new model is introduced. In addition, large teams of developers may choose to use multiple programming languages, including Microsoft Visual Basic or Visual C++, or NI LabVIEW. It is imperative that the software components of an integrated measurement system today are easily maintained and can be rapidly modified.
Integration with Enterprise Tools

With the growing popularity of corporate and worldwide networks, measurement and control systems can realize numerous advantages. Developers can use networks to distribute information instantaneously, update manufacturing processes continuously, and update product designs on a regular basis.

Often, developers must distribute automated measurement and control systems across a manufacturing floor, through several different buildings, or across the globe. Worldwide corporate database systems must easily share information collected with one of these systems. The development tools used to build automated measurement systems must be tightly integrated with the enterprise tools on which companies are standardizing. Without this integration, organizations face a severe loss in productivity and competitive advantage, as time to market increases rather than decreases.

The Importance of a Measurement and Automation Software Framework

The challenges that system developers face today lead to the need for an integrated software framework. This framework must decrease the complexities of integrating multiple measurement devices into a single system by providing standard interfaces to all I/O devices, and must provide development tools to rapidly configure, build, deploy, maintain, and modify high-performance, low-cost measurement and control solutions. This integrated software framework must provide seamless connectivity to the ever-evolving enterprise management systems on which an organization is standardizing. It is through this framework that an organization delivers products to market faster, achieves greater product quality, and lowers development and production costs.

An integrated Measurement and Automation Software Framework delivers a modular, yet integrated, structure for building high-performance, automated measurement and control systems. For maximum performance, ease of development, and system level coordination, the components of the framework must be independent, yet tightly integrated (see Figure 4). This modular, integrated structure empowers developers to rapidly build measurement systems and modify them easily as the system requirements change.

Developing a measurement and control system with a tightly integrated software framework delivers numerous benefits, including:

- **Significantly increased** productivity throughout the development, deployment, maintenance, and modification process with rapid application development tools designed for measurement and control applications
- **Higher performance** measurement and control systems, as the tools at each level are designed to work well together to deliver maximum measurement and control performance
- **More tightly integrated systems** that bring together numerous diverse measurement devices into high-level systems that connect easily to other processes throughout the organization
- **Decreased costs** throughout the product life cycle
With these benefits, organizations become more competitive because they can design and test higher quality products and deliver them to market faster and more cost-effectively than ever before.

For maximum benefit, a Measurement and Automation Software Framework must include the following:

- Measurement and Control Services Software that seamlessly connect to numerous I/O devices and provide high-level interfaces for simplified system development
- Application development environments that tightly integrate with both the Measurement and Control Services Software and system management services
- System management services to organize data, tests, and high-level systems

**Measurement and Control Services Software**

The Measurement and Control Services Software plays a critical role in delivering the key benefits of a modular computer and networked-based measurement system. The components of this software – hardware drivers, flexible high-level application programming interfaces (APIs), and a configuration manager – must all integrate within the application development environments (ADEs) to attain maximum system performance and development productivity. The specific tasks of the Measurement and Control Services Software include integration of measurement devices, as well as local and distributed configuration and programming of the measurement devices.
Important Attributes of Measurement and Control Services Software

Too often, developers of measurement and automation systems assume the existence of a device driver alone is sufficient for integrating their measurement device. The device driver should offer the key benefits of fast performance, device programming flexibility, a consistent and scalable API, local and remote configuration and operation, and a seamless integration with the ADE. In the ideal implementation of the Measurement and Automation Software Framework, the software that controls the measurement devices is transparent, appearing only as part of the ADE. This ideal implementation guarantees maximum flexibility in development and a scalable architecture that organizations can deploy on all of the platforms targeted by the ADE.

Figure 5 shows a typical implementation of the Measurement and Control Services Software. In this system, I/O services control message-based devices (or traditional instruments) connected by GPIB, serial, VXI, USB, 1394, and Ethernet interfaces. The I/O services for these devices are delivered in the NI-488.2™ and NI-VISA™ software drivers. The NI-DAQ™ software drivers control the high-speed electronic measurement devices. NI-DAQ controls the high-performance, modular data acquisition and control hardware installed directly in the computer or connected by USB, 1394, or Ethernet. Similar to NI-DAQ, NI-IMAQ™ controls image acquisition devices, and NI-Motion™ operates motion controllers. These software services share a common integrating framework for channel expansion, real-time synchronization, fast programming, high-speed streaming, and remote device access.

Integrating Traditional Instruments

Many measurement systems continue to integrate traditional instruments from a variety of vendors. The NI-488.2 and NI-VISA drivers provide hardware independence that protects users from time-consuming modifications to source code when equipment needs to be changed. With NI-488, users can migrate their GPIB instruments from a PC to a network or from Windows to Linux to an embedded real-time OS.
NI VISA, a virtual instrumentation software architecture standard for instrument control, provides a layer of hardware independence so that engineers can quickly benefit from the technological advances of the PC and the Internet. NI-VISA abstracts user code from the physical interface between the computer and the instrument. Whether the instrument uses GPIB, Serial, VXI, Ethernet, 1394, USB – or any future technology the PC and instrumentation industry might adopt as mainstream – instrument control software written to the NI-VISA standard works seamlessly, facilitating code reuse as instrument control standards evolve. Instrument drivers built on top of NI-488 or NI-VISA exploit these benefits and deliver additional productivity by incorporating instrument knowledge within the instrument driver itself, which greatly simplifies program development.

**Maximizing Performance and Development Flexibility**

Measurement devices integrated via the NI-DAQ, NI-IMAQ, and NI-Motion software driver architectures offer maximum performance and development flexibility. The key features of these drivers are:

- Maximum I/O performance for measurement devices by controlling registers directly on the devices and transferring data directly into and out of computer memory
- Device driver architecture that conforms completely to all operating system interoperability standards (e.g. Windows 2000, Windows XP, Mac OS)
- Scalable APIs that offer programming simplicity and scalability across embedded, desktop, and distributed systems
- Seamless integration with ADEs such as LabVIEW, Visual Basic, and Visual C++
- Internal driver architecture that permits distributed operation of devices connected via USB, 1394, and Ethernet
- System integration of measurement devices with synchronized timing and triggering using the RTSI™ bus architecture for PCI and PXI measurement devices
- High-speed streaming of data to maximize measurement system throughput

**A Common Configuration Framework**

The configuration manager, known as Measurement & Automation Explorer, presents a unified system view of measurement hardware supported in the Measurement and Control Services Software. With Measurement & Automation Explorer, users can define channel names to organize signals or specify scaling functions to convert digitized signals to measurement quantities. The key benefit of the configuration manager is the integration with the ADEs, such as NI LabVIEW and LabWindows/CVI, as well as NI Measurement Studio, which provides measurement components for Microsoft Visual Basic. This integration gives developers the ability to easily integrate multiple measurements into a single application without tedious programming. Without these configuration tools, developers must spend needless time configuring these measurement functions programmatically.
Application Development Environments

ADEs play a critical, visible role in a Measurement and Automation Software Framework. With these tools, the system developer designs and integrates the system that takes measurements, controls processes, displays information to the end user, connects with other applications, and much more. Just as a home computer user today is more intimately knowledgeable about their Web browser than their modem card, it is in the ADE that system developers spend the majority of their development time. Not only must ADEs integrate tightly with Measurement and Control Services Software as well as enterprise-wide data and test management systems, but also it is imperative that the ADE provide high-level, intuitive development paradigms so that a wide variety of users can rapidly build measurement and control systems.

Important Attributes of an ADE

Too often, developers of measurement and automation systems assume that they can successfully use any programming language to combine components into an integrated system. They often feel that they should choose a tool based solely on the maximum amount of flexibility it offers. At the opposite end of the spectrum, some developers opt to build their tools in-house or use turnkey tools that work only with a single measurement device. Both of these options can severely limit productivity and prevent a measurement and automation system from achieving maximum performance, development productivity, and integration with other systems.

The ADEs used to develop measurement and automation applications should provide an easy-to-use design model, compiled performance, and application-level programming flexibility for a complete range of applications. Equally as important, these ADEs should tightly integrate with Measurement and Control Services Software that connect to a wide variety of I/O devices, as well as with system-level tools such as enterprise database systems or test management systems into which applications can integrate either now or in the future.

Commercial, Off-the-Shelf

Despite the perceived flexibility benefits of using in-house built software architectures, organizations that use proprietary software often incur unintended expenses as they fight to keep up with rapidly advancing technologies, such as operating systems and networking technologies, that are not core to their business. This effort can siphon valuable resources away from business operations and often lead to a loss of valuable time. For example, by using an off-the-shelf ADE designed for measurement and automation, developers can quickly and easily upgrade to the latest operating system, or integrate emerging Internet and XML standards with minimal development investment. Or, when using an ADE such as Visual Basic or Visual C++, specialized measurement and automation add-ins can significantly reduce development time.
Ease of Use

Because the ADE is where systems are assembled, ease of use in these tools is critical to the productivity of system developers. Although it is an important element, ease of use goes beyond how quickly someone can get up and running. With easy-to-use ADEs, developers can easily integrate processing routines with multiple measurement devices, create sophisticated user interfaces, deploy and maintain an application, and modify the application as product designs evolve and system needs expand.

Integrated Measurement-Specific Functionality

In addition to the tight integration with Measurement and Control Services Software, it is critical that the ADE used to develop a measurement and automation system can seamlessly manage and process measurements. To do this most effectively, the ADE should incorporate measurement data types directly in the environment so that these measurements are easy to use in additional processing routines. For maximum development productivity, the ADE should include comprehensive statistical and numerical analysis functions, as well as high-performance signal processing and control algorithms common in measurement applications. ADEs should integrate typical routines found in measurement and automation applications, including functions such as PID and fuzzy logic control, noise reduction, spectral measurements, digital filtering, response measurements, signal detection, numerical integration and differentiation, curve fitting, fractional-octave analysis, and order analysis.

Scalable

By standardizing on a common, scalable measurement and automation ADE within an organization, the organization saves administrative costs, builds internal expertise, and streamlines development and system interoperability. Productivity increases by standardizing on one suite of tools for multiple user types and measurement and control applications. For example, the following scenarios could all use the same software application:

- **Research laboratory** (flexibility is key) – acquires and displays data from a digital multimeter (DMM) and oscilloscope connected via GPIB to a desktop computer
- **Field service application** (portability is key) – acquires and displays data from a computer-based DMM and oscilloscope plugged into a laptop via PCMCIA
- **Production test application** (speed and cost are key) – acquires and displays data from a computer-based PXI/CompactPCI DMM and oscilloscope plugged into a PXI chassis

Tight Integration to Enterprise Systems

The ADE must provide open connectivity to other software tools in the enterprise. This open framework ensures that developers can integrate their measurement systems with applications throughout the enterprise – by integrating with other software tools through ActiveX and DLLs, connecting to corporate databases through standard structured query language (SQL) and application database object (ADO) interfaces, connecting to the corporate network through technologies like TCP/IP or UDP, and creating shared reports through formats like XML and HTML. As Microsoft.NET technologies drive tighter connectivity between remote systems, this requirement will only grow in importance.
NI ADEs for Measurement and Automation

NI offers two ADEs specifically for measurement and automation – LabVIEW and LabWindows/CVI. In addition, NI Measurement Studio features measurement and automation add-ins for Microsoft Visual Basic and Visual C++.

With NI LabVIEW, an off-the-shelf graphical development environment designed specifically for developing integrated measurement and automation systems, developers assemble user interfaces and high-level functions for data acquisition and control, signal processing and analysis, and visualization in the same way that flowcharts are constructed. With the modularity and hierarchical structure of LabVIEW, users can quickly and easily prototype, design, deploy, and modify systems. Because the LabVIEW ADE is compiled for maximum execution performance, contains hundreds of measurement-specific analysis routines, and is tightly integrated with the Measurement and Control Services, developers can quickly design high-performance measurement and automation applications that incorporate numerous I/O devices. LabVIEW also integrates well with industry-standard Internet and database technologies, as well as test management systems.

NI LabWindows/CVI is a proven powerful ANSI C development environment with comprehensive acquisition, analysis, and presentation libraries designed to protect your C-based development effort for years to come. With its development productivity features including drag and drop user interface design, interactive creation and testing of code, and complete instrument driver creation wizards, LabWindows/CVI simplifies the task of creating C-based measurement applications. First introduced in 1988, LabWindows/CVI has proved a solid and reliable development environment that has withstood the many changes in operating system and computing technologies. LabWindows/CVI integrates tightly with the Measurement and Control Services to integrate a wide variety of I/O.

NI Measurement Studio provides an extensive collection of easy-to-use instrument control, data acquisition, signal processing, and visualization tools for use with Microsoft Visual Basic and Visual C++. This functionality is exposed through ActiveX controls and class libraries, a measurement application wizard, and custom ActiveX controls for user interface creation. The components of Measurement Studio tightly integrate with the Measurement and Control Services to provide maximum development productivity to system developers. In addition, the Measurement Studio tools tightly integrate with test management software and easily incorporate with database and Internet applications.
System Management Software also plays an important role in the Measurement and Automation Software Framework (see Figure 6). These tools, while not required for some simple measurement applications, are invaluable in large, integrated systems.

Two common system management environments include those for managing large amounts of data, as well as those that manage large test systems. In both cases, it is important that the tools used for these systems tightly integrate with the ADEs and Measurement and Control Services Software. As with ADEs, commercial, off-the-shelf system management tools can save organizations countless hours compared to systems designed and maintained in-house. In addition, these environments must integrate well with other systems in the organization, including databases and Manufacturing Execution Systems (MESs).

**Important Attributes of Test Management Software**

Test management software, commonly referred to as a test executive, must provide a framework for the entire test system. It must include sequencing, looping, and decision-making capabilities to the test program, generate reports, and provide an interface between the test system and various enterprise systems, such as databases, manufacturing execution systems, and quality systems.

In the past, the test executive framework often resided inside the actual ADE-built tests as they executed. The software was intertwined and impossible to program separately. This meant that maintenance and support was difficult and code reuse was minimal for all but the simplest projects. More recently, the test code, or “step,” and the test executive framework are defined and abstracted along broadly functional lines. Test executives today have evolved to automatically identify the UUT, undertake all of the sequence control, and carry out test result evaluation, reporting, and user management.
With commercial, off-the-shelf test management software, developers can create custom test solutions that merge an out-of-the box solution with a full range of customization options to deliver test systems cheaper, faster, and with more features.

NI developed TestStand™, a ready-to-run test executive that organizes, controls, and executes automated prototype, validation, or production test systems. TestStand is completely customizable, so users can modify and enhance it to match their specific needs. TestStand fully integrates with LabVIEW, LabWindows/CVI, and Measurement Studio so users can generate code and perform full debugging, including stepping into test programs directly from TestStand.

**Important Attributes of Data Management Frameworks**

If a measurement system collects large amounts of data, data management frameworks are an important element to that integrated measurement system. By incorporating technical data management into a system, organizations can share data efficiently and make informed decisions. Data management software provides a framework for measurement systems to store data in an organized repository so that users can retrieve the data from numerous file formats, manipulate and interactively analyze the data using measurement-focused routines, and generate standardized professional reports for collaboration.

By implementing a data management framework and using it with NI DIAdem™ software, users can extract measurement data from numerous file formats, manipulate and interactively analyze data using hundreds of engineering-focused analysis routines, and generate standardized professional reports for collaboration.

**Implementing an Integrated Measurement and Automation Software Framework in an Organization**

National Instruments has been a leader in measurement and automation software for the past two decades. From its Measurement and Control Services Software to its revolutionary ADEs, the LabVIEW graphical development environment and LabWindows/CVI ANSI C development environment and Measurement Studio for Microsoft Visual Basic and Visual C++, to its software platforms for test management and data management, National Instruments offers a tightly integrated software framework to increase productivity throughout the design and deployment of automated test, measurement, and control systems.

With commercial, off-the-shelf technologies, developers gain more productivity in their organization by focusing more time on application needs and less time on rearchitecting software for upcoming technology changes. By taking advantage of NI Measurement and Control Services Software, they can easily move from one hardware architecture to the next without rewriting code. With software that scales from the simplest application to the most challenging, and from the design lab to the manufacturing test floor, organizations realize significant productivity gains as they standardize on a single suite of tools within their organization, build expertise within the organization, and facilitate the reuse of measurement systems from one functional area to the next.
By capitalizing on the National Instruments Measurement and Automation Software Framework to build integrated high-performance, low-cost measurement and automation systems, organizations become more productive, save significant development time, and take advantage of a platform that can expand and evolve as needs change for many years to come. This, in turn, decreases time to market and increases competitive advantage, as higher quality products can be designed and tested more quickly than ever before.

To help organizations get started, NI provides on-site consultation, start-up assistance, and numerous training alternatives, from self-paced interactive CDs to on-site training courses. In addition, fully trained Application Engineers can assist customers during system development. If a developer wishes to benefit from this integrated software framework, but does not have in-house development resources, NI partners with more than 600 companies worldwide through the NI Alliance Program. These companies are experts in the NI software framework, and can work with customers to deliver tightly integrated measurement and automation applications that exactly meet their needs.

To start lowering development and production costs, achieving greater product quality, and getting products to market faster, call your local NI office for a free consultation with an expert. Visit ni.com/products for information on specific software products in the Measurement and Automation Software Framework.