Is there "one" controller that can do it all?

Bob Nelson, Controller Marketing Manager, Siemens Energy & Automation, Inc.

www.sea.siemens.com

SIEMENS

Automation Controller

September 1, 2008

What kind of controller is best? Is there "one" controller that can do it all? You have many options, including the PLC (programmable logic controller) and the PAC (programmable automation controller). But when it comes down to it, the name is not what's important. What is important is developing a clear understanding of your needs and then finding the controller that best targets them. It is critical to base a controller decision on the requirements of the entire automation system... no matter what kind of TLA (three letter acronym) is on the label.

This paper addresses what you can expect from a modern, state-of-the-art automation controller. It discusses how the controller can help solve a variety of application challenges, including:

- Logic control
- Motion control
- Flexible architecture
- Process control
- Safety
- Redundancy

Beyond this, you must consider the software tools required to support all of these challenges to realize a true multi-domain controller platform.

White Paper

What kind of controller is best for your application? Is it a PLC (Programmable Logic Controller)? Perhaps you should use a PAC (Programmable Automation Controller)? Maybe a PAD (Programmable Automation Device) would be best? When it comes down to it, the name is not that important. What is important is developing a clear understanding of your needs and then finding the controller that best targets them.

In the world of industrial automation, technology has evolved leading to innovations in controllers and I/O, enhanced engineering tools, and entirely new system architectures. The controller we are familiar with can now take virtually any size and shape, from a traditional industrial PLC, to a "soft" PCbased controller and even a "nano-sized" brick.

Industrial automation as we think of it today began with the development of the PLC 40 years ago. The relay ladder logic program was only a few hundred bytes in length, looked just like a typical electrical drawing, and could be programmed and maintained by most plant electricians. Since then, program size has ballooned into the mega-bytes; programming tools have become more sophisticated; network communications have evolved to connect "islands" or "silos" of automation into plant-wide networks; and the PLC has grown from being a dedicated box running sequential ladder logic to an open platform for controlling applications across all manufacturing environments. The automation project has evolved into a full-fledged software development project, needing structured design, extensive diagnostics and alternative programming languages to achieve project goals.

By 2002, the evolution of the PLC was so dramatic that Craig Resnick, Director of Research at technology consulting firm ARC Advisory Group, proposed changing the name of the PLC. "The label 'PLC' simply understates the capability of current automation systems," wrote Resnick, proposing a new name: "Programmable Automation Controller," or PAC. He noted that "the PLC has evolved by incorporating open standard interfaces, multi-domain functionality, distributed modular architectures, and modern software capabilities integrated as turnkey automation solutions."

Walt Boyes, editor in chief of CONTROL magazine, goes on to state "Today, everyone – at least the major companies who make full-functioned PLCs - are making PACs... even if they don't call them PACs. There is no advantage in buying something called a PLC or a PAC; you just buy the controller you need." Regardless of the three letter acronym used, Craig Resnick's bottom-line recommendation was that "end users should let their application be their guide" in choosing control solutions. If your application should be your guide, let's look at how an automation controller can help you meet the requirements of your application.

What should you expect of an automation controller? One thing is clear: the modern automation controller can do more than the "founding fathers" ever imagined. A single controller can now do the job of multiple dedicated controllers of the past. Considering the advances in technology, a user has every reason to expect that a single controller deliver value through:

- Providing cost-effective and best-of-breed performance across all desired control applications
- Employing open standards to allow full integration in a heterogeneous, multi-vendor control environment
- Offering upgrade and maintenance paths to ensure long-term flexibility, protection against obsolescence and a favorable total cost of ownership
- Delivering sustainable, long-term suppler relationships based on decades of experience and global technology leadership

On paper, an automation controller must cover a broad range of application requirements. The platform should offer standards-based openness; flexible, single-database integration for the entire system; standard IEC-61131-3 control programming and configuration languages; the ability to create preconfigured libraries of reusable code; and objectoriented design of complete system architectures.

The reality is that very few controllers can truly provide these capabilities in one package. But such solutions do exist, and offer both the openness suggested by the "PAC" promise and the best-of-breed features that address the real-world solutions users need to satisfy requirements, present and future, across multiple types of applications.

Six Key Control Applications

What does it mean when you hear that modern automation controllers offer multi-domain control capability that can be used across application types? One way to look at this is to identify unique control applications, and then look at what the controller should do to help you solve the application. Following is a list of six unique control applications:

- Logic control
- Motion control
- Flexible architecture
- Process control
- Safety
- Redundancy

If we look at each of these independently, and identify some core elements that you must have, we can start to paint a picture of what the modern automation controller must deliver.

Logic control

- **Performance / processing speed** In machine, factory or process control, controller performance is a key element in productivity gains. As you strive to improve the electro-mechanical performance of your application to yield higher throughputs, the performance of your automation system must keep pace. The execution time, high-speed interrupt handling, and segmentable scan times of your controller all contribute to the performance of your application.
- One program across a variety of controller platforms (scalability)

Ideally, you would have one engineering environment to configure all of your applications, which, in turn can be downloaded to the target platform meeting your requirements. This separation of program and platform allows you to focus on solving the application problem, followed by selecting the best, most cost-effective platform for your specific application.

- Configurable system-wide diagnostics You can always program diagnostics into your code, but this takes time, and often is sacrificed to meet project schedules. A controller with built-in diagnostics that can be easily enabled or configured ensures that this important aspect of your solution won't be skipped.
- Availability of multiple programming languages Relay Ladder Logic is expected in a logic controller, but aspects of your application may be better implemented using graphical function blocks or a high-level programming language. The IEC 61131-3 standard for PLC programming languages defines five languages that range from ultra-efficient "machine code" to the graphical representation of sequences. Compliance with this standard not only gives you

implementation options, but can also ease training across vendor platforms.

• Potential for reusable code

The idea to create, test, and reuse common functions is embraced by programmers in all industries. The goal is to be more efficient and to improve program quality. For industrial automation, leveraging a library (from the supplier or defined by you) of common program elements and using these elements repeatedly reduces implementation time and improves program consistency, maintainability and quality.

• Investment security

Rather than demanding new products, many users push for longer life-cycles for their existing products and architectures while taking advantage of advances in technology. The result is having the "latest and greatest" without the need to "rip and replace."

Motion Control

September 1, 2008

- Tight integration of motion and automation
 The reality of machine control is that both motion
 control and automation must work in tandem
 nobody likes the task of integrating between
 dedicated control devices. Not only is manual
 integration time consuming but it is prone to errors
 and maintenance challenges. Ideally, motion and
 automation engineering would be done in the same
 software with the same languages.
- High-speed, deterministic performance Motion applications tend to be very fast and therefore require realtime, high-performance controllers. However, speed is nothing if it's not repeatable. For proper operation, the controller needs to execute your defined tasks the same way every time. This is defined by determinism, and measured by jitter. Realtime indicates that a process or action happens in a defined time. Determinism is where this response time is "fixed" and does not deviate significantly. Jitter is a measure of the deviation from the defined time. Optimally, you want fast, deterministic execution with very small jitter of all elements of the motion solution from controller to network to drive to the feedback mechanism.
- The controller should be designed to make it easy to engineer motion applications The ability to propagate libraries and templates throughout the application is very important to minimize rework and promote the use of standards.

The use of standards-based, pre-defined, tested functions saves significant time. Access to standard algorithms (like circular interpolation and 3D movement) frees you to focus on configuring your application rather than programming base motion functionality.

• Tools to optimize performance

Motion applications tend to be very fast. To optimize performance, tuning and trace tools are absolute requirements. These tools must look at the entire motion solution to perform accurate tuning. Beyond visualization, the tuning tools may provide wizards to guide you to the best dynamic parameters.

 Reliable and established networks to support the motion controller

Proprietary networks tend to limit you to only one supplier. Ideally your motion controller supports an established network standard. It is critical that the network also accommodates high-speed deterministic motion applications. The network should not introduce any limits to the performance of the motion solution – the network should be transparent.

Flexible Architecture

 Improved integration to achieve higher performance and lower engineering costs
 Frequently, the ideal application solution may require specific hardware and custom software technology to tightly integrate with your controller.
 Using a traditional PLC architecture this could involve costly engineering integration along with performance limiting communication interfaces. An open architecture makes it possible to embed the custom program directly with the automation control program. Additionally, you can achieve high-performance due to the ability to directly plug interface modules into the platform bus.

When an application is outside the capabilities of classical controller architecture, open architectures are essential. Connecting nonstandard communication busses, PCI-bus based servo controllers, specialized vision applications, and applications written in non-PLC programming languages are not easily done in a classical architecture. Automation platforms utilizing an open architecture allow the user to develop their own integration solutions that are run as part of the controller execution.

Another example is when the output of a machine

is so customized that it must change very often (possibly for each product produced). For this, it is not realistic to rely on human intervention to change the parameters or to load a new recipe. In this instance, a direct connection to a database is preferred. In a traditional controller solution, this usually involves complex network communications, middleware applications, database interfaces, etc. A controller using an open architecture allows for direct connections to the database, possibly even on the same platform.

- The ability to think beyond typical restrictions How often have you wished for more memory or processing power? Traditional controller platforms limit you to specified resources – if you need more, you must purchase a different platform. Automation platforms based on an open architecture provide the flexibility to add additional memory, or even to take advantage of faster CPUs.
- Leverage continuous platform innovation Hopefully, we are all familiar with Moore's Law. Intel's Gordon Moore has proven to be a visionary in predicting the pace of innovation for computing. Computer processor speed directly impacts controller program performance in an open architecture controller. Coupled with continuous performance improvements to the controller engine and its associated peripherals and the result is dramatic improvements in overall performance. A traditional controller can not offer the ability to leverage this rapid innovation.

Process Control

- Availability of pre-engineered solutions, templates, and extensive libraries
 Process users look to leverage pre-defined objects from libraries (i.e., cascade PID) rather than "rolling their own." This speeds up configuration and makes the resulting project more consistent and maintainable. It would be advantageous if you could also create your own library objects based on your unique control strategy.
- Ability to select program execution speed The normal reaction to controller speed is "faster is better." However, for process applications regulatory control loops normally scan in the 100 to 500 millisecond range. In some cases, it could be detrimental to have control logic execute any faster – possibly causing excessive wear on final control elements such as valves, resulting in premature maintenance and process issues. It is important to

have the ability to select program execution speed.

• Emphasis on design using a top-down approach to engineering

Process users spend a lot of time on up-front design and overall program structure. This focus on upfront design minimizes costs, compresses project schedules and creates applications that can be maintained by plant personnel over the long term. Since many process applications are large and plant-wide in scope, the ability to propagate libraries and templates throughout the application is very important to minimize rework and promote the use of standards.

• Ability to be installed in a hostile environment Many process applications are in hazardous (i.e., explosive) environments. Additionally, the environment may be moist or corrosive, potentially leading to damage of critical electronics. The automation platform and associated peripherals must withstand such environments without undue installation costs or complexity.

Embedded knowledge

Technology is just part of the challenge to produce an effective automation platform for process applications. It helps tremendously if the supplier knows process control and has embedded this knowledge in their technology and the available libraries. The result is a platform designed to meet the unique requirements of process applications "out of the box."

Safety

 The use of a single controller for both standard and safety functionality

When your application requires both standard automation and machine safety, you can install parallel automation and safety systems. However, eliminating the need for two controllers saves significant cost and reduces complexity. The optimal solution is a single controller that handles both tasks. This opens new possibilities when designing architectures such as a shared controller, shared bus system and shared I/O serving both standard automation and the safety.

• One engineering platform to program standard as well as safety logic

Learning engineering environments takes time you may not have. A controller that uses the same engineering tools for standard automation and machine safety configuration not only simplifies learning the tool, but dramatically simplifies the integration of diagnostic information. The same diagnostic visualization you already use for your automation system can be used by maintenance and operations to troubleshoot safety incidents.

 Safety as part of a distributed architecture, even with existing systems

Implementing programmable safety on top of an established automation system is a challenge. Rather than putting in place a parallel, safety-only system, you should consider whether the safety controller can be used as a slave to the established automation system, handling the safety functions. Doing so has a significant impact on reducing integration time and effort.

Redundancy

• Uninterrupted control of a process or machine It may sound obvious, but redundancy implies a backup system – a safety net to protect your process from unexpected controller failures. Do you want controller redundancy only, or do you want to take redundancy to other levels? What about the power to the controller, or the networks you rely on, or the inputs and outputs connecting your process? Not only must you determine the depth of redundancy you require, but you need to determine how your application behaves during switchover. Is a switchover time measured in seconds okay, or do you need milliseconds? The automation platform should be able to meet your requirements, regardless of your answers to the preceding questions.

Minimize engineering complexity

Implementing redundancy should be a simple task. If you take on the challenge of designing your own redundancy solution, you must make sure you've considered all modes of failure and how the system responds. You also need to consider keeping the primary and backup controllers synchronized.

A better approach may be to take advantage of redundancy from the supplier, whether embedded in the operating system of the controller or implemented in software (for less critical applications). The important thing is to let the automation platform take care of redundancy for you while you focus on solving your application problem.

• Quickly resolve a failure

When you have a failure, your redundancy strategy will keep your process or machine running. But how do you know that a failure has occurred, and how do you fix the problem? This is where diagnostic notification comes into play pinpointing the problem. Once you locate the problem, it is very beneficial to be able to hot-swap the failed components. If the failed component happens to contain a program, is the program automatically loaded, or does this require manual action from you? All of this combined serves to minimize the time to repair a failure, getting you back to a redundant state.

• The value of the product being manufactured and the cost of downtime

Consider the value of your product and the true cost of downtime. If the value of the product being manufactured is relatively low and/or downtime results in lost production but little additional cost or damage to the process, implementing redundancy may not be the best choice. If the value of the product is high (either in raw material cost or market value) and downtime not only results in lost production but potentially dangerous and damaging conditions, the nod should probably go to full redundancy. In process applications running 24/7/365, downtime is one of the gremlins you try to avoid at all cost. The more volatile the application, the more it may require a solution with lots of redundancy.

Moving beyond hardware – engineering software is needed to realize your solution

With so much expected from a single controller platform, the engineering tools to support all the disciplines involved in a typical automation system become more important then the controller itself. A true multi-domain controller platform includes the multidisciplinary tools required to support traditional logic applications, as well as, motion control, process control, and all the other applications previously discussed.

Defining the controller's "personality"

A complete suite of engineering software gives you the tools to manage the entire engineering life cycle – from design on through ongoing maintenance. Capabilities you should look for include:

- An integrated engineering environment for logic, motion, process, etc.
- A single point for system-wide engineering with a project view
- A modular structured approach where you can design your automation around your process
- The flexibility provided by support for the IEC 61131-3 languages

Today's best-of-class engineering software includes programming and configuration, advanced diagnostics, simulation, remote maintenance and plant documentation – for the controller and other system components like HMI and drives. It also allows for the development of interfaces to nonstandard or industry-specific devices.

Which 'One' Controller Is Right For You?

Manufacturers and machinery builders need to know how the automation controller can solve their specific needs while adapting to future requirements. The best controller is the one that most closely enables your automation system to meet the requirements of your manufacturing application, and allows your manufacturing process to integrate seamlessly with other manufacturing and business processes. It is important to base a controller decision on the requirements of the entire automation system... no matter what kind of TLA (three-letter acronym) is on the label.

Bob Nelson, Controller Marketing Manager for Siemens Energy & Automation has over 20 years experience in the automation and control industry. The author can be reached at: bob.nelson@ siemens.com

www.sea.siemens.com

All rights reserved. All trademarks use are owned by Siemens or their respective owners.

© 2008 Siemens Energy & Automation, Inc.

Siemens Energy & Automation, Inc. Industry Sector Drive Technologies and Industry Automation 3333 Old Milton Parkway Alpharetta, GA 30005 Printed in USA

A white paper issued by Siemens. ©2008 Siemens Energy & Automation, Inc. All rights reserved.