

InTech

www.isa.org/intech

with **Industrial** Computing®



DREAM COME TRUE OR NIGHTMARE?

By William Hodson

Process control companies did, after all, recognize the value in distributing control.

Distributed control systems (DCS) protect their business interests by dragging their feet and not fully embracing the open high-speed protocols that are now available say some consultants, detractors, and critics.

Is it that simple?

Focus on the issues involved with use of a truly open protocol including interoperability assurance, performance, robustness, determinism, capacity, integrity, security, responsibility, and more.

Is openness a dream come true? Alternatively, does it open Pandora's Box for the user?

The vendors did start it all

The first industrial process control highways were

conceived and designed by process control companies as they recognized the value in distributing control and the associated need to be able to communicate with its various distributed functioning nodes.

The communications needed to be fast, extremely reliable, secure, and environmentally hardened. The services needed to support specific functions including alarm reporting, rapid data gathering, parameter display and modification services, downloading of code, and downloading of databases.

Nothing available on the commercial market came close to serving those requirements. Therefore, the DCS vendors invented their own, generally called *proprietary* protocols.

Sometime during the late 1970s or early 1980s, The Institute of Electrical and Electronics Engineers (IEEE) developed a token passing highway usable in process control, IEEE 802.4.

The ISA's SP72 Committee created the *PROWAY* standard based on 802.4, but it gained little acceptance.

An attempt by General Motors to create a Manufacturing Automation Protocol in the manufacturing arena based on 802.4 also had so few followers that it ultimately failed.

Although 802.4 had some followers, with the lack of full application protocol services, it had no real hope of assuring any degree of interoperability.

Finally in 1985, the ISA chartered the SP50 Committee to create a fieldbus communications standard for instruments and actuators. The ensuing "Fieldbus Wars" resulted in an International Electrotechnical Commission (IEC) standard that contains eight non-interoperable specifications.

Meanwhile, out in the real world, the commercialization of the combined Ethernet and Transmission Control Protocol/Internet Protocol (TCP/IP) physical, data link, network, and transport layers created an inexpensive communications basis over which to send messages. To no surprise, many already-existing application protocols merely found a way to hop on top of the Ethernet plus TCP/IP lower layers, but applications were still far from being interoperable.

Today, although there is a leading candidate for a high-speed Ethernet-based process-control-related communications protocol, there has not been widespread acceptance of it as an infrastructure. Why is that? What are the issues?

Far outperforms commercial

Like many open protocols, there are limitations. One such limit is performance. Such protocols cannot usually contend with the extreme measures necessary to provide for large systems needs. They are generally simple.

For example, a basic data interconnection mechanism is for one device to publish one variable at a time in one message. These single-variable messages then rout, one at a time, to their destination. Communications engineers can assure there is a large overhead per message. Ethernet requires a 64 byte message as a minimum (to insure proper recovery), while publications are typically little more than a dozen bytes long. So what happens to performance in 10,000

or 50,000-point systems?

Be mindful that with smart protocols, usually come more measurements because the incremental cost is lower for multivariable devices sharing a single interface and common bus. This drives measurement point counts up, toughening the challenge for a given application.

While a recent protocol extension permits multiple variables to be contained in a single publication from a common source, such an extension is insufficient to fully address the concern.

There are DCS manufacturers' protocols, which share some or many of the open protocol features that do not rely on single variable publications. They provide interfaces between their high-speed infrastructure and the lower-speed H1 links, the fieldbus interface module.

At least one of these interfaces, unlike its open *linking device* counterparts, does NOT merely republish messages. Because of the need to interface two networks that vary by more than three orders of magnitude in performance, and the need to service a large numbers of nodes handling large amounts of data, the fieldbus interface module has a complex patented design. It has a database of data relationships. The source of that database is the vendor-provided device-descriptor files. This technique lets the interface determine and use the most efficient technique to read the data in bulk,

"Bottom line, a propriety fieldbus interface module and optimized protocol can outperform open protocols and can allow a DCS vendor to provide users with rapid, state-of-the-art performance on even the largest of installations, whereas fully open architectures may not."

cache it, maintain a refreshed cache for future predicted access, and manage the cache to avoid unwanted H1 bus activities.

It far outperforms commercial, off-the-shelf linking device designs based merely on the open specifications. However, be forewarned that the advanced fieldbus interface device looks the same as the simple one on the network diagram; there is nothing diagrammatically apparent to reveal the performance differences.

Further, instead of merely routing the Foundation fieldbus single message protocol over a high-speed infrastructure, some DCSs use an optimized protocol.

Such a protocol uses larger message structures and powerful gather-scatter techniques to efficiently distribute real-time data among the nodes in a robust manner. It provides reliable transfer of data, even on an exception basis, and provides periodic updates of lists of subscribed data elements.

Therefore, bottom line, a propriety fieldbus interface module and optimized protocol can outperform open protocols and can allow a DCS vendor to provide users with rapid, state-of-the-

art performance on even the largest of installations, whereas fully open architectures may not.

A staggering number of tests

Some well-established DCS manufacturers provide extensive performance, capacity, and topology testing for every update and every release. That means they support certain defined and supported topologies and repeatedly test (qualify) those topologies under stress. Another expectation of users is the DCS vendors will support various combinations of old and new equipment. It is not uncommon for a user to continue operation of a controller and its I/O for two decades or more. This legacy support requires purchasing and maintaining numerous vintage versions of controllers, hardware, firmware, and software, as well as adding to the numerous combinations of regressions testing.

Considering the complexity of functions and possible combinations of functions performed by a controller and its I/O, the number of tests can be staggering. A qualified test engineer will perform a large number of tests based on a judicious selection of regression tests that represent an appropriate set of tests. This may rest on code coverage, knowledge of what has changed, areas deemed most sensitive, most critical areas, and a selective set of generalized tests.

In order to assure users proper operation, various configurations of the system test up to the limits of designed capacity, assuring at least minimum performance specifications are still attained and maintained.

This proves to be a very expensive overhead in terms of labor hours, expansive capital equipment, installation costs, and maintenance costs. There is no direct revenue derived from these tests. Test plans are developed, reviewed, executed, and the resulting documentation archives to an extensive database associated with a release of the system.

DCS vendors manage all this to assure that equipment meets customers' demands and as part of customer care. If users want the ability to merely add any number of manufacturer's non-validated products anywhere in the system, they should be prepared to design, perform, and record similar tests or must prepare plant management to accept the risks of their alterations to the configured and validated control system and the risks to plant production.

New techniques to gain access

Beyond performance, capacity, and topology, a few DCS vendors provide extensive interoperability testing. Although not-for-profit interoperability organizations provide registration and testing services, we have found that subtle interoperability problems may remain.

We have worked at length with other vendors, even direct competitors, to be mutually reassured that our systems and products and their systems and products interoperate as expected, to the users' benefit. Tests have proven to go well beyond those of the registration process.

This testing extends to other open products including power supplies and network switches. In one very noteworthy finding, we learned a firmware upgrade to a previously released product caused severe performance degradations.

We work with vendors to identify and resolve such concerns. Users building their own system with open components may share similar experiences.

Most computer users have some experience with the devastating effects that a virus, worm, or other ill-intended executables can inflict upon their personal computer. Since most *upper* ends of DCSs today platform on pervasive Microsoft technology, the opportunity to inflict damage on a control and monitoring system is real.

Terms and definitions

Distributed control system (DCS):

A system of dividing plant or process control into several areas of responsibility, each managed by its own controller (processor). The whole interconnects to form a single entity, by using communication buses of various kinds.

Fieldbus interface module: A shared electronic circuit between two functional circuits or devices. This might be the boundary between different fieldbuses or between fieldbuses and field devices.

H1: A fieldbus network that operates at 31.25 kbit per second. Generally, it is a lower-speed and lower-cost network.

Overhead (per message): In communications, overhead is all the information, such as control, routing, and error-checking characters, used in addition to the transmitted of interest. Overhead includes routing information, operational instructions, and retransmissions of data that arrive at the endpoint containing errors.

Regression testing: A selective retesting of a software system that has just had modifications. It is to ensure that any bugs are now fixed, that no other previously working functions have failed because of the reparations, and that newly added features have not created problems with previous versions of the software. Regression testing takes place after a programmer has attempted to fix a recognized problem or has added source code to a program that may have inadvertently introduced errors.

Most of us are aware that Microsoft operating systems have come out with security vulnerabilities, which even include remote access susceptibility. However, a simple denial-of-service incident can result from something as simple as an uncontrolled overload. Such peril may be accidental or deliberate.

It might be from a disgruntled employee or ex-employee, or it may be sabotage from a competitor, foreign terrorist, mentally deranged individual, or a hacking student who thinks causing undesired operation is challenging and fun. However, whatever the source, the consequences to a process plant could be grave in terms of life, injury, equipment damage, or production loss.

While most common forms of destruction arrive via e-mails that the user must open, incidents that are more recent have revealed new techniques to gain access to a control system that do not require such user cooperation.

With an open operating system, more users understand the details and weaknesses of the operating environment. Hence, more potential knowledgeable individuals might use their knowledge to become a threat.

Such infiltration may be simply to render it inoperative. Alternatively, its purpose may be to cause a deliberate form of disoperation or upset intended to harm equipment, employees, or even neighboring inhabitants. DCS vendors concern themselves with system designs to minimize the possibility of such attacks, including load throttling.

If one intends to build a *do-it-yourself* system using fully open components, consideration for such possibilities needs to be a part of the plans and budget.

Error and failure detection

Responsible DCS vendors address robustness of critical functions within the system. Failure modes and effects analyses can highlight areas where failures can have most significant effects.

By considering the severity of the failure, probability of occurrence, ability to detect the failure, and actions needed to address the failure, one can prioritize the solutions to produce the greatest degree of robustness possible for the effort.

The majority of the system's logic is for error and failure detection and recovery, versus the primary function of the system. Do-it-yourself system builders need to perform these failure modes

and effects analyses per their system design and each time their system changes. Consultants can help less experienced users.

Despite the increasing power of microprocessors, the increasing speed of networks, and the best-intended efforts of interoperability organizations, distributed control systems can be vulnerable to faults and overload conditions if non-validated products connect in an open manner. A DCS is a tightly integrated set of modules with distributed functions. Dependable DCS vendors strive to protect their users by limiting access, qualifying third party devices, executing extensive interoperation, performance, capacity, and topology testing, insuring robustness, and providing guidelines for acceptable loading, interoperation, and security of various products. Periodic releases of the system also assure qualification of interoperation of the defined set of products supported.

Users who plan to create their own fully open systems need to plan to commit capital for full-capacity equipment to be dedicated to testing. They need to staff up with trained testing engineers who can develop thorough and selective battery of test scenarios to establish confidence in capacity, performance, interoperation, and security. Automated testing with automated recording and scoring can help to reduce the costs.

True openness does not come without a price. Along with the ability to connect anyone's anything anywhere anytime comes the risk that untested elements may not reliably function as intended, may interfere with the proper operation of other elements, and costly unpredictably loss of plant production may result. In order to assure the "dream come true" does not become a "nightmare," those risks can best be mitigated by comprehensive, methodical testing and re-testing in large-scale system configurations as new elements and upgrades to existing ones are considered for use.

Behind the byline

William Hodson has been a member of ISA since 1969 and is an ISA fellow. He is an engineering fellow with Honeywell in Fort Washington, PA, has two degrees in electrical engineering, and is a licensed professional engineer. Hodson is a senior member of IEEE and has over 35 years of experience in the field of digital process control. His e-mail is william.r.hodson@honeywell.com.

Reprinted with permission from **InTech**, March 2005.

© 2005 ISA Services, Inc. All Rights Reserved. FosteReprints 866-879-9144