Re-engineering the distributed control system

FRED P. BADER

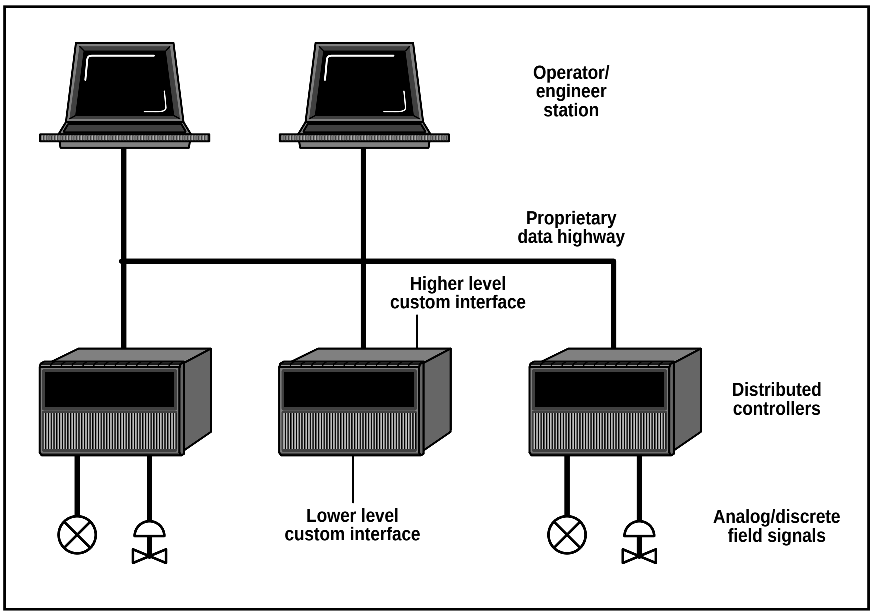
The proprietary distributed control system (DCS) is being challenged and changed by a host of new initiatives and standards that are turning the DCS into a network of diverse components.

Distributed control systems (DCSs) are the architecture of choice for most of today's process control systems, and many versions of the DCS are available. They have undergone constant change since their introduction, achieving new levels of performance, operational flexibility, and capacity as new microprocessor technology and software tools have become available. Throughout all the changes one factor has remained fairly constant: proprietary implementation of control software, distributed communications, and human machine interface (HMI) software—all implemented across supplier-specific hardware platforms.

When computer performance and memory capacity were limited, the primary design challenge of all DCS suppliers was to develop a secure integrated control system. Any attempt to utilize standards or technology developed for other (larger) markets adversely affected both performance and security. As the DCS market grew, user investment in training and support of specific systems created both real and psychological barriers to accept change. Innovation, performance improvements, and cost reduction that occurred in most other industries applying computers did not occur in the DCS. Consequently, the DCS has become known for its high cost.

Today's microprocessor performance and capacity, software standards, communication standards, and operating systems are no longer barriers to achieving low-cost DCS performance and security. It is now possible to implement "component"-style DCS, and activities in the marketplace tend to support the transition to this concept. Perhaps this concept is the real meaning of "open" systems, a term that has been confused and abused for the last several years.

1. Proprietary DCS



Traditional DCS

Figure 1 depicts a classical form of today's DCS. It consists of clusters of distributed "unit" controllers communicating along a redundant network to a set of operator consoles. Field signals are connected by individual wire pairs to input/output (I/O) cards. (The field devices are intelligent in many cases but do not alter the basic architecture.) The system is proprietary. All the control software, communications software, and HMI software are implementations of proprietary functional requirements. With complete single-source control over the functions, an integrated environment was developed. Well defined control and operator response, along with secure data transfers along the network, was assured. Access to the systems from outside applications was handled by interface modules furnished by the DCS supplier. Since the number of outside applications and niche products is very extensive, a custom software interface needs to be developed for each unique application or device. It is also recognized that no one supplier can furnish all of the best potential applications, leading to a need for many custom interfaces. This has led to a need to find better solutions for interfaces. As a result, one of the changes occurring in the DCS is the transition towards standards and open technology that allows these applications or products to be more easily assimilated.

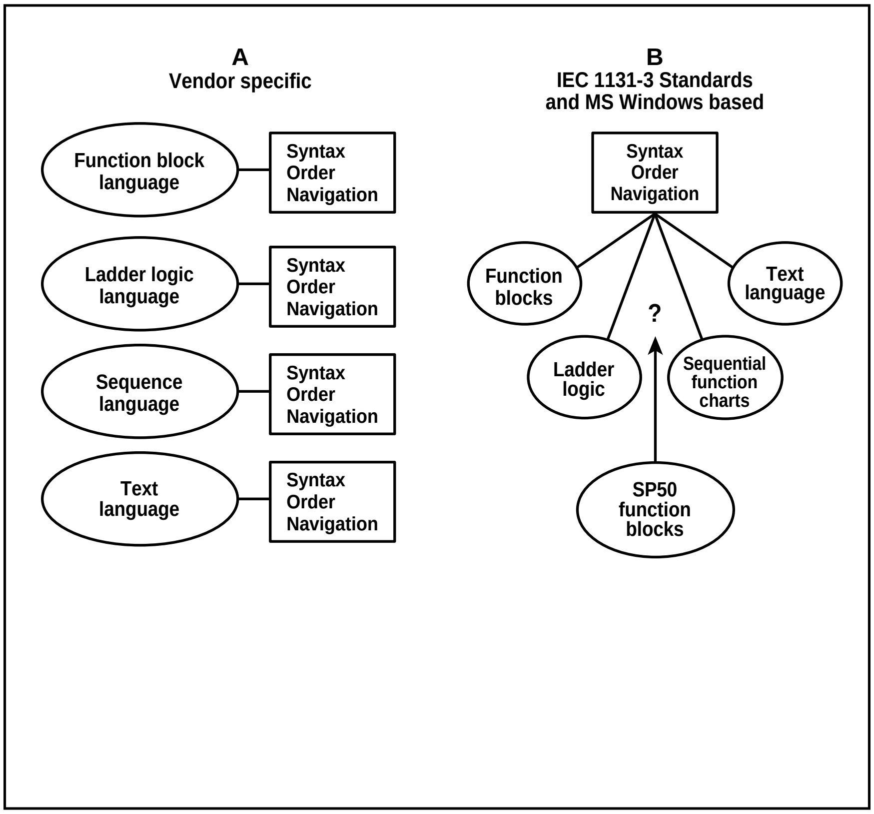
Proprietary to standard control software

Each DCS version has been characterized by its novel and unique methods to implement the control program. No standard ever existed in this area. As shown in Fig. 2A, each DCS implementation incorporated its own control software language sets, plus the attendant unique syntax, navigation, and order. An experienced DCS professional could adapt to these idiosyncrasies because all the implementations were fundamentally achieving the same result. To most users, however, it is difficult to make an easy transition from one system to another.

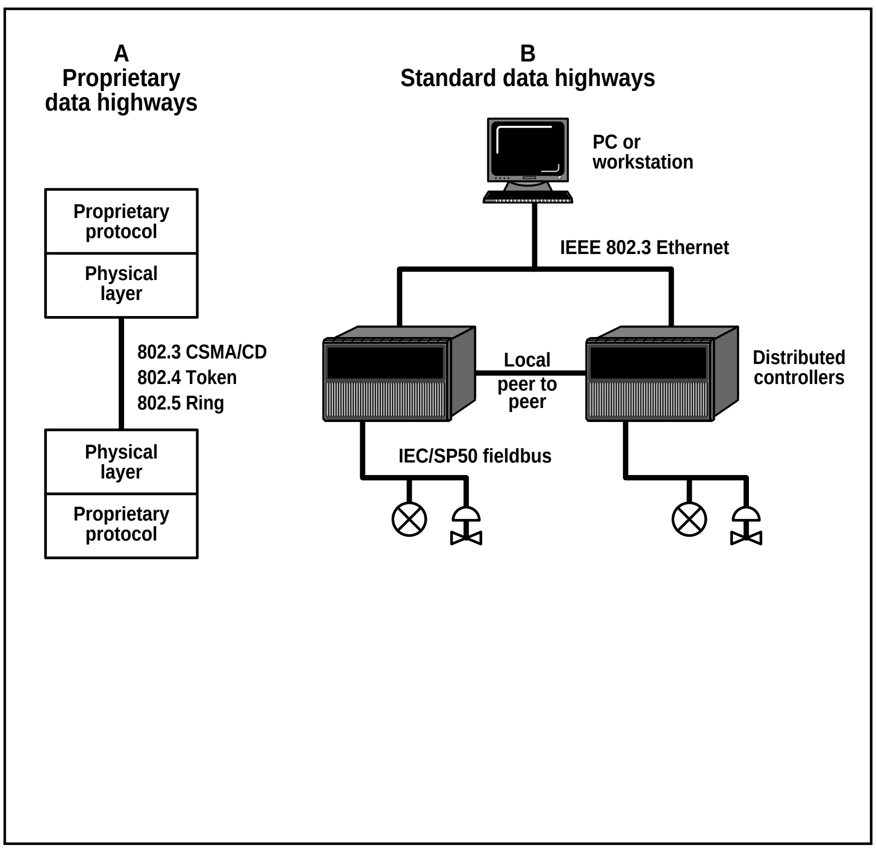
Figure 2B depicts the beneficial change to control software created by standards. IEC 1131-3, an international software standard, uses a common syntax, navigation, and order for up to five defined languages. As a standard, IEC 1131-3 can be implemented by any supplier and thus is of great benefit to users seeking better integration at this level.

Other control language standards are eminent. Efforts to promote fieldbus acceptance incorporates a common function block set to permit devices to operate with each other and to be interchangeable. Many users rightfully challenge the benefits of fieldbus in their plant or mill operations (over existing smart transmitter techniques) but recognize the benefits from a common set of control algorithms. The IEC 1131-3 standard

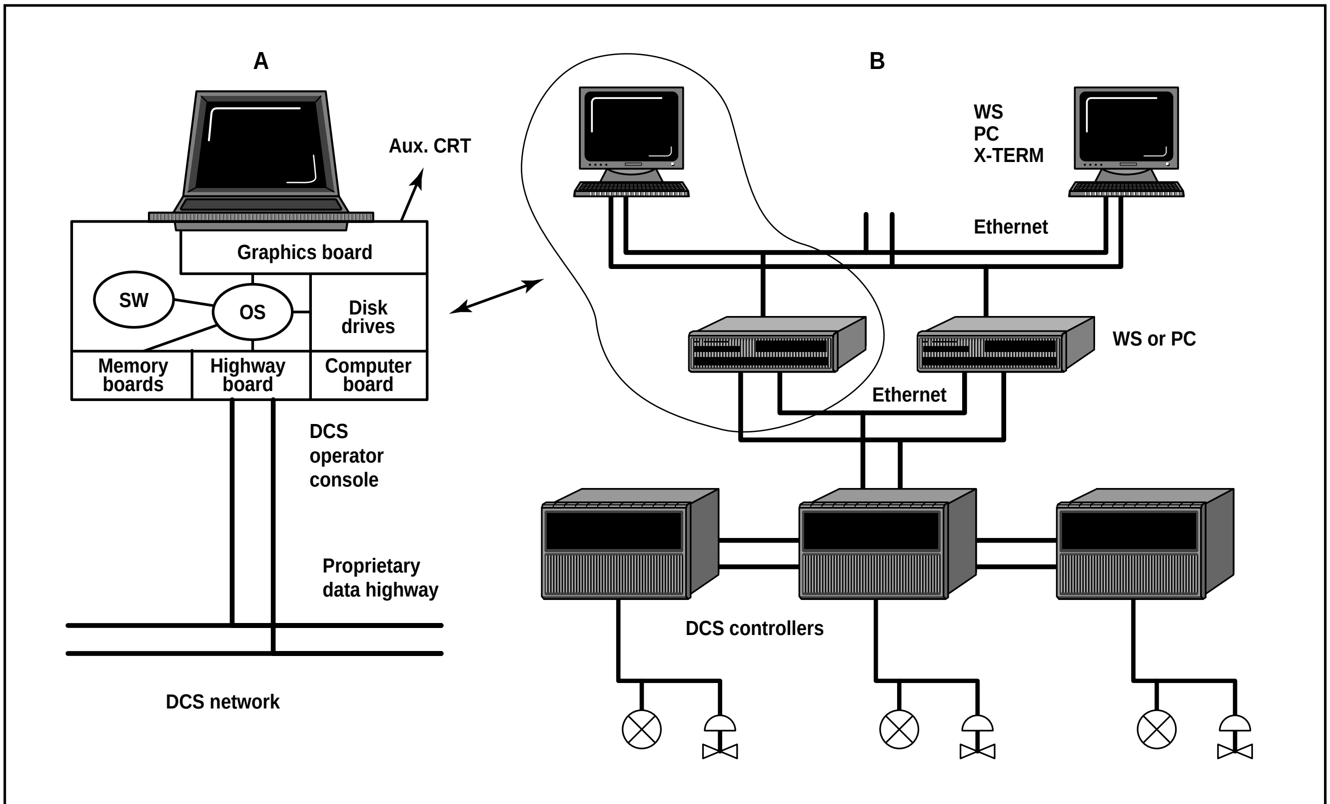
2. Proprietary to standard control software



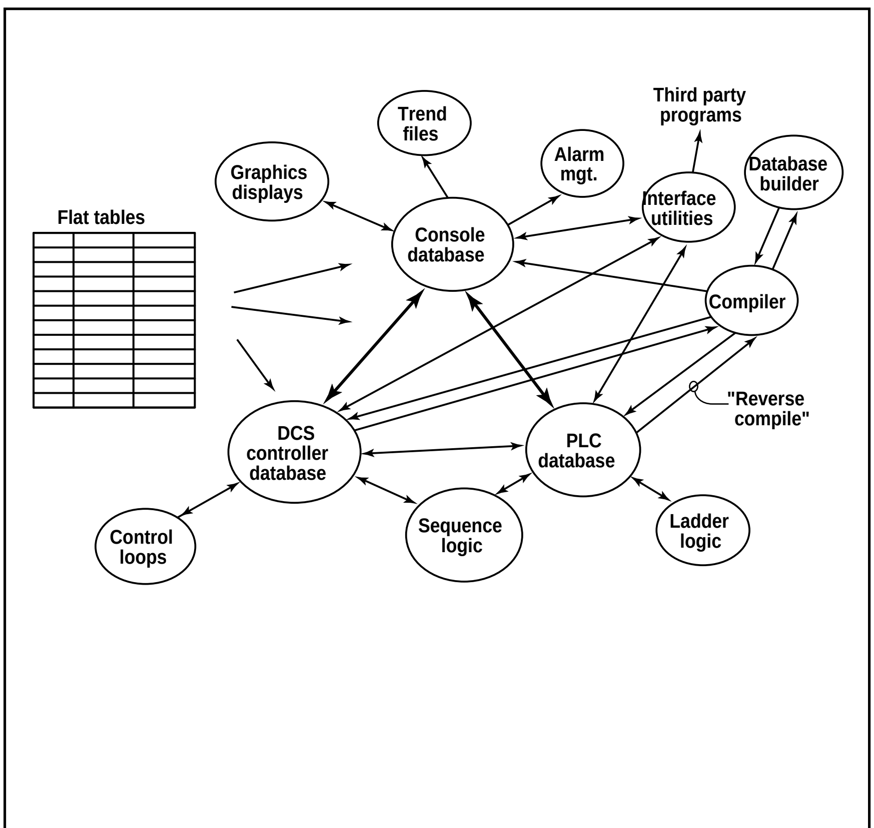
3. Proprietary to standard communications



4. Proprietary to standard operator station



5. Typical DCS software integration



also incorporates a standard control algorithm set and can be adapted for fieldbus use. In fact, the Fieldbus Foundation has publicly recognized the existence of the standard, has indicated a willingness to work within the International Electrotechnical Commission (IEC), and has acknowledged IEC 1131-3 as part of the overall fieldbus standards initiative.

Proprietary to standard communications

The classical DCS borrows some of its communication technology from the computer market and other existing standards. In Fig. 3A, the DCS distributed elements communicate along some form of bus or ring topology defined by Institute of Electrical and Electronic Engineers (IEEE) standards (802.3, 802.4, 802.5). The physical layer is standard, but the communications protocol is proprietary. The protocol implementation was always enhanced to accommodate redundancy and security. The enhanced protocol rarely was accessible or documented for use by external applications. Early efforts to achieve market acceptance of true open protocols in communications were not successful due to high node cost and lack of availability in the computer industry. The most obvious example of this was MAP (Manufacturing Automation Protocol).

With little process industry objection, a communication standard has emerged as a defacto standard. IEEE 802.3 Ethernet with TCP/IP protocol has become widely accepted. Included as a standard offering on all UNIX workstations, and widely available on any PC via a plug-in card, Ethernet has become too obvious to ignore for use in process control systems. Dual-channel architectures provide a form of redundancy, and restricted, isolated access produces data security and performance. Lower-level peer-to-peer controller networks can still pass critical control data, but external accessibility at the Ethernet level provides a cornucopia of open systems applications not available in any proprietary DCS.

6. Open software integration

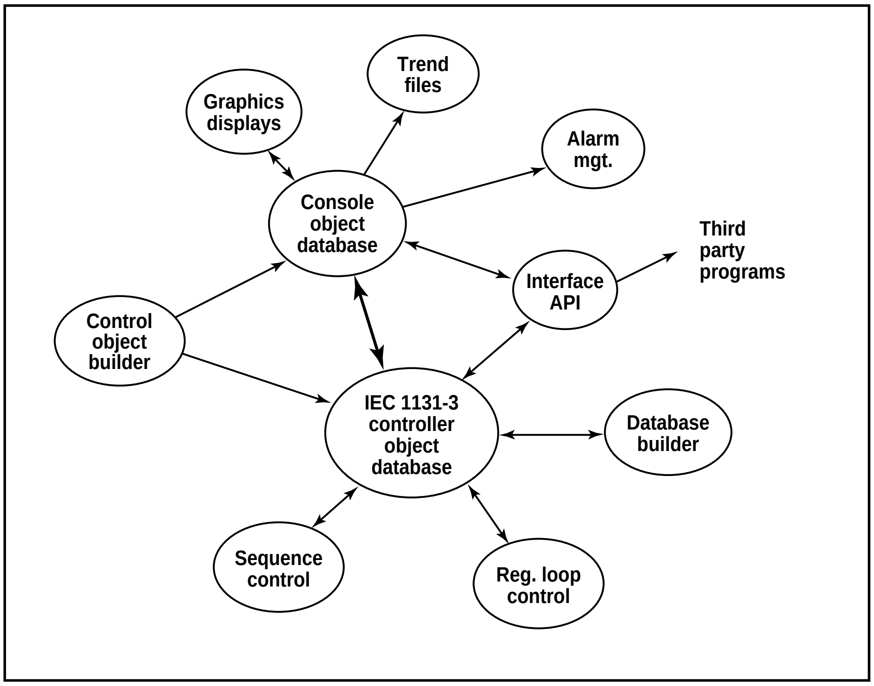


Figure 3B depicts the open DCS system Ethernet communications layer between the distributed controllers and operator workstations, essentially creating an open communications layer at the plant production/operations level. On the horizon is fieldbus, the open communication system for smart field devices, and Asynchronous Transfer Mode (ATM). ATM will produce even greater performance improvements as the replacement for Ethernet TCP/IP in the future. A number of open contenders for the fieldbus exists today (e.g., Hart, FIP, Profibus, DeviceNet, Arcnet, Lonworks) and ISA/IEC standards work continues. Eventually, a single standard will emerge as the communications network below the control computers. When the fieldbus is combined with the IEC control software standards and the TCP/IP Ethernet communications standard, the DCS does appear to become an open control system up to the operator console. At this level, other computer industry forces are at work to change forever the concept of the operator station.

Proprietary to standard operator stations

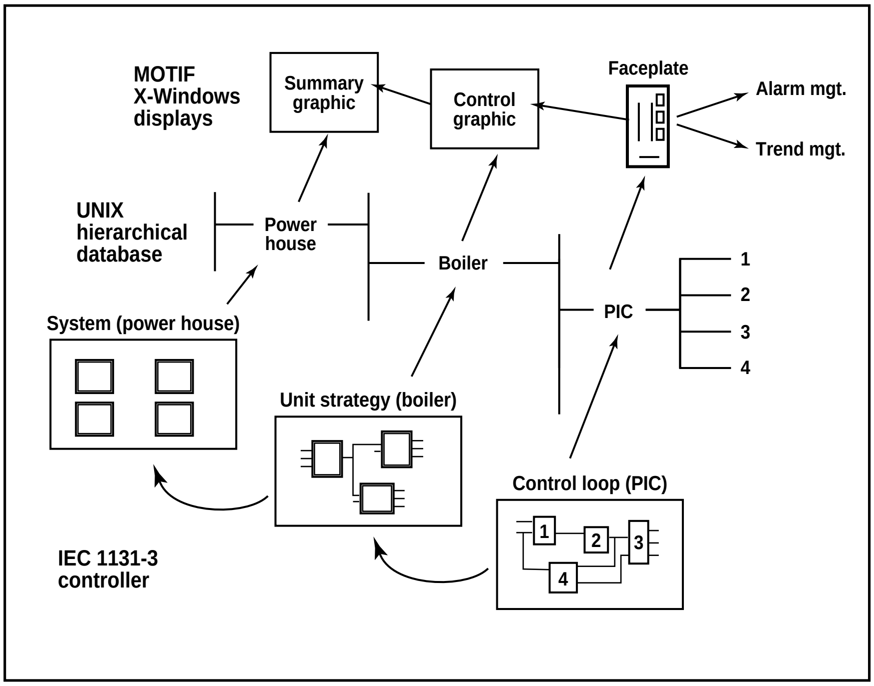
The DCS operator station has always been a mix of vendor-specific and commercial computer offerings. In some systems, a minicomputer or commercial workstation has been the only operator console. This is not surprising, since the operator station most closely resembles a multitasking minicomputer in functional requirements. Tasks such as operator interaction, alarm management, trend history, report generation, and graphics builders are readily accomplished on a multitasking operating system. For small systems, PCs readily meet the requirements. As the cost of PCs and workstations plummeted, and demand for better business systems integration occurred, it became obvious that DCS suppliers would have to yield to this market change. Performance, operating systems, software tools, capacity, graphics resolution, and communications, as developed for other industries, all are viable in a DCS—and at low cost. The change to standard commercial platforms was inevitable.

Figure 4 shows the difference between the DCS console of the past and the workstation of the future. In Fig. 4A, an operator console consisted of a collection of PC boards and peripherals (e.g., disk drives) consolidated into a vendor-specific industrial package. In Fig. 4B, a commercial workstation replaces the whole ensemble. There is still a need to provide a rugged display station for an operator in some instances, such as on a paper machine or batch plant floor location. Industrial terminals provide this function while using the same standards for communication and interaction as the other DCS components.

DCS software integration

Perhaps the most significant advantage of the proprietary DCS was the database integration between the distributed components. Suppliers have worked to minimize database duplication efforts and provide secure communications and validation of the distributed database. The primary methodology used in DCS systems is the correlation of the controller database and the operator console database, as shown in Fig. 5. Lists or tables comprising the controller database are passed to the console as points or tags to create the console database. Also included on most legacy systems are substantial PLC systems with their own databases that must be merged into the DCS.

7. Open system object data base integration



From this database, applications such as dynamic graphics displays, trend history, alarm management, reports, etc. are spawned. In an effort to provide better data management support, interfaces to popular database programs and computer-aided design packages were developed to assist in the design and documentation of the systems. Unique interfaces existed across all the different systems to allow third-party business, plant information, and production systems to access needed process control data. Since the basic DCS database was proprietary, it became necessary to create custom interfaces to support each external application (such as the PLC interface) at significant initial and long-term maintenance cost.

In most DCSs today, the controller database is a compiled form of a unique higher-level language implementation (e.g., function block programming). Since the controller is distributed, most controllers have a local configuration panel or utility to allow plant or mill personnel to make quick changes to maintain production. This capability is somewhat self-defeating, since it becomes difficult to update and maintain the original system database. Elaborate schemes to “reverse compile” the local controller database to the system database have had some limited success, but the only guaranteed method to avoid database mismatch is to keep the high-level form of the database in the controller. Technology today makes this concept a practical reality.

Open software integration

With the advent of standards and with defacto commercial hardware and operating systems invading the DCS at all levels, the issue of software integration of distributed databases becomes obvious. Fortunately, new software tools and methodologies have emerged concurrently that not only make integration possible but also make it easier to attain.

DCS control strategies have always been a collection of software modules or routines (e.g., function block) combined together to fulfill a basic application (e.g., single loop with alarms). Groups of modules produced multivariable control schemes (e.g., heater control), and the multivariable schemes were grouped to provide a unit control system (e.g., digester, boiler, column). Units were further combined into process plants (e.g., pulp mill, power plant, acid plant). This three-dimensional view of the software can be created by object programming techniques (a term to define a software algorithm with its procedure and data as an independent object) and hierarchical database structures (as opposed to flat database tables). The IEC 1131-3 Control Programming Standard is conducive to both of these software practices. The standard can be implemented by object programming of all its multilanguage elements, and its derived block structure creates a hierarchical database. Early DCS controllers were unable to mimic this real-world process environment. Because of processor performance and memory limitations, most controllers resolved their control strategies into compressed binary tables for execution. Today’s processors and memory capacity, of course, can meet this more complete object structure.

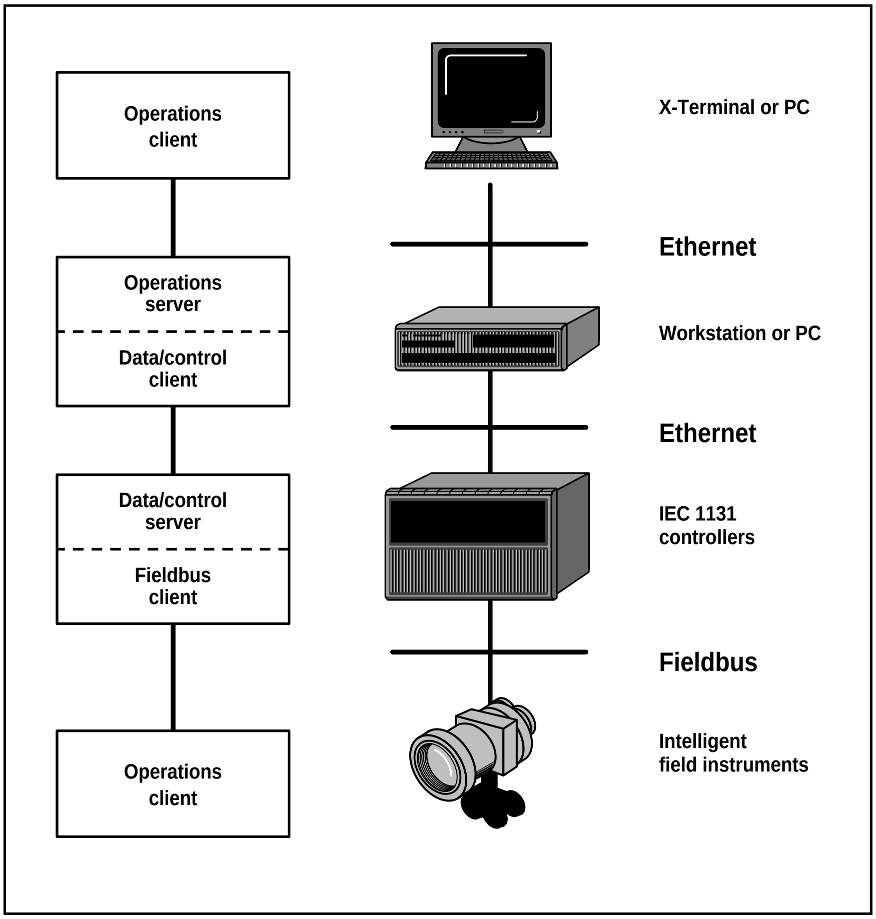
With this IEC 1131-3 object hierarchy existing as the foundation of distributed control, faithfully reproducing the control database at the operator level is a progressive step. As shown in Fig. 6, connecting object databases together between the controller and consoles is more efficient when the object, with all its attributes, can be copied in a single command. Database generation is fast and more secure, since process control objects can be prebuilt for an entire class of process control applications traditional in the industry (e.g., loop controllers, ratio controllers, motor controllers, indicators). The IEC 1131-3 standard also produces a significant benefit to DCS system database management. The traditional method of separating DCS and PLC functions into two dissimilar products is no longer necessary. The IEC 1131-3 controller is comfortable serving both needs. Also, with the ability to store the graphical control strategy in the controller, it is no longer necessary to reverse compile programs to maintain the database.

Hierarchical database connectivity

Methods and technologies exist today to make the hierarchical database connection quickly between multivendor offerings as part of open system progress. Diverse suppliers are producing products that address the common problem of integrated database connectivity with little or no collaboration with each other. This indicates that the supplier industry is recognizing the problems and producing attractive solutions.

Figure 7 depicts a multisupplier solution that copies hierarchical object databases from the controller to an open UNIX or PC console. In the figure, function block objects representing IEC 1131 setpoint blocks, PID blocks, auto/manual output blocks and alarm blocks are collected into a pressure control (PIC) object. The PIC is a subset of a boiler control strategy, and is incorporated into an IEC resource as a powerhouse controller. At the console, a hierarchial database convention awaits an application instance to create a specific operator console. By assigning an HMI (human machine interface) reference within the control strategy loop sheets, the HMI station immediately copies the object and HMI attributes into the HMI database. The necessary boiler control scheme database becomes a part of the console database. In this multisupplier database integration, diverse hardware and software have been integrated. The controller is an industrial controller from one vendor implementing the IEC 1131-3 software standard, the operator station is a UNIX workstation from another vendor, and the workstation HMI software database is from still another vendor. The hierarchical database structure specified by IEC 1131-3 is faithfully reproduced at the HMI level to provide an integrated DCS-style solution that embodies the real plant layout.

8. Client/server DCS systems



Client/Server DCS

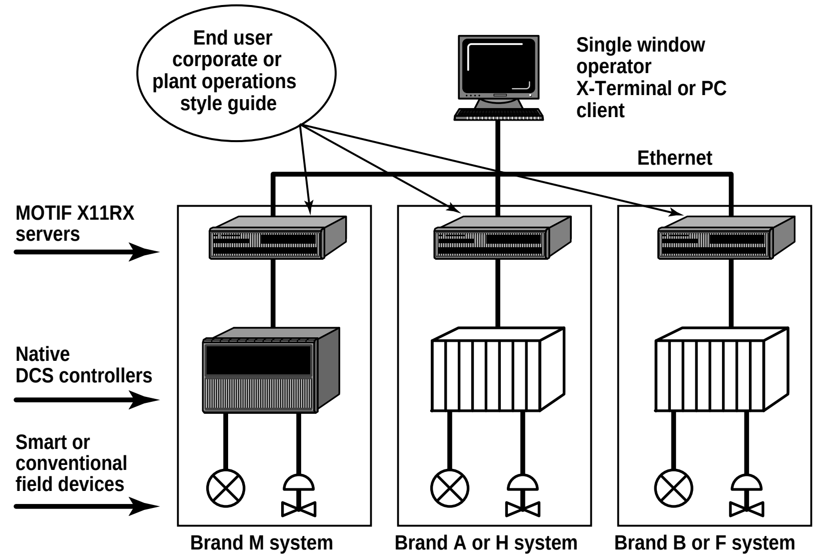
As the concept of client/server computing has become more accepted and widespread, the open re-engineered DCS uses this technology to great advantage. A client/server relationship exists across the DCS system. In Fig. 8, fieldbus devices (e.g., transmitters) act as “servers” of information to a controller “client.” After processing all its local control needs, the controllers act as a server of information to a workstation client. This client incorporates a database plus a set of operator applications (e.g., graphics displays). To provide these applications to a user CRT screen, the workstation acts as a server to display station clients requesting the data. These display station clients can be other workstations, X-terminals, or personal computers.

Throughout the entire chain of cascaded client/servers, from the field instruments to the operator, a set of international or defacto standards has been used to perform system integration. In the past, the overhead of “open” systems was performance and cost prohibitive, and DCS systems were engineered as proprietary solutions to get the proper performance. But today’s processor speeds and memory capacity make such open integrated interactions possible. With the open infrastructure now in place, it is possible to achieve significant user benefits in DCS component selection, cost reduction, and operational uniformity.

Multivendor system integration

If all vendors are driving their systems towards “open” solutions and connectivity, it stands to reason that the systems should become interoperable. Certainly, this is one of the major benefits purported for fieldbus. One area where standards and open systems can provide significant user benefit is in plant operations and production. Most process industry companies would readily admit their primary goal is to make their products at the highest uniform quality and lowest cost. Operating or production departments are challenged daily to find the best methods to achieve these goals. Unfortunately, no one supplier has all the best tools and products to satisfy this need. If operations had a uniform method to interact with all the various systems (control, advanced control, information,

9. Multi-vendor systems integration



production, and lab) across the entire plant or organization, regardless of the supplier, the resultant savings and operating efficiencies would be dramatic. It would also be desirable to achieve this potential without sacrificing competitive selection of technological innovation.

This vision is possible with today's technology. In Fig. 9, a single (or multiple) X-window station can interact with any number of servers of any system supporting the X11RXMOTIF standard. Each server could in fact be furnished by separate suppliers. Left as is, each supplier provides its own interpretation of what is best for users in the form of operator displays, display navigation, point details, faceplates, graphics symbols, alarm presentation and management, trend presentation and management, report styles, and systems diagnostic displays. From an operational standpoint, little has been accomplished with the new technology, since the production departments across an enterprise must deal with all the idiosyncrasies of each unique offering. What is suggested here is a corporate user standard operations methodology. By defining a company "style guide to operations," a user company can define a CRT operations standard that best fits its needs. Certainly, the user has the best overall expertise about how to run their plants or mills. The style guide, installed in the respective servers of the automation and production systems, produces a common "look and feel" at the HMI X-terminals, regardless of the connected control or production system. Any plant personnel involved in operations, production, maintenance, and management would benefit from the style guide. It is recognized that differences exist in graphics generation packages in separate HMI products. However, the MOTIF navigation style is standard (and used by all MOTIF compliant suppliers). Elements such as point details, faceplates, display hierarchies, alarms management, trend manipulation, report layout, etc. can be uniformly defined in a style guide by the user for the benefit of all the plant operators.

Such a scenario implies a "standard," leading some to the conclusion that an industry organization should define the style guide. With so many different software offerings and tools currently available, and the need for users to protect internal production methods and processes, such a conclusion could be misguided. Individual user companies should create their own corporate standards.

Conclusions

Hardware and software technology changes will continue to evolve and progress as long as there is a competitive environment for suppliers to find markets. At the DCS level, these changes are rewriting the concept of a DCS from a proprietary single offering of older technology into a fast-paced world of component solutions. The conclusions that result from these changes are:

- Lower cost, component DCS architectures
- Easier and quicker integration
- More selection of best fit hardware/software tools
- Greater use of low-cost commercial technology
- Implementation of more standards in software and communications
- More interoperability
- Corporate or plant DCS operations standards. TJ

Bader is director of strategic alliances, Moore Products Co., Sumneytown Pike, Spring House, PA 19477-0900.