Volume EMC 04.01

ISA Encyclopedia of Measurement and Control

Networks for Factory Automation

Samuel M. Herb, Author

- Nuts and Bolts of Networking
- Physical Structures
- Logical Structures
- Open Communication Standards
- Digital Field Communications
- Architectural Issues
- Connecting the Enterprise
- Plant Information

Taken from the book: ***Understanding Distributed Processor Systems for Control***

ISA is the international society for measurement and control®

Notice

The information presented in this publication is for the general education of the reader. Because neither the authors nor the publisher have any control over the use of the information by the reader, both the authors and the publisher disclaim any and all liability of any kind arising out of such use. The reader is expected to exercise sound professional judgment in using any of the information presented in a particular application.

Additionally, neither the authors nor the publisher have investigated or considered the effect of any patents on the ability of the reader to use any of the information in a particular application. The reader is responsible for reviewing any possible patents that may affect any particular use of the information presented.

Any references to commercial products in the work are cited as examples only. Neither the authors nor the publisher endorse any referenced commercial product. Any trademarks or tradenames referenced belong to the respective owner of the mark or name. Neither the authors nor the publisher make any representation regarding the availability of any referenced commercial product at any time. The manufacturer's instructions on use of any commercial product must be followed at all times, even if in conflict with the information in this publication.

Printed in the United States of America.

ISA
67 Alexander Drive
P.O. Box 12277
Research Triangle Park
North Carolina 27709

Editor's Introduction

This "mini-book" is available both in downloadable form, as part of the ISA Encyclopedia of Measurement and Control, and bound in a print format.

"Mini-books" are small, unified volumes, from 25 to 100 pages long, drawn from the ISA catalog of reference and technical books. ISA makes mini-books available to readers who need narrowly focused information on particular subjects rather than a broad-ranging text that provides an overview of the entire subject. Each provides the most recent version of the material—in some cases including revisions that have not yet been incorporated in the larger parent volume. Each has been re-indexed and renumbered so it can be used independently of the parent volume. Other mini-books on related subjects are available.

The material in this mini-book was drawn from the following ISA titles:

- *Understanding Distributed Processor Systems for Control*, by Samuel M. Herb, Chapters 22-31. Order Number: 1-55617-645-7

To order: Internet: www.isa.org
Phone: 919/549-8411
Fax: 919/549-8288
Email: info@isa.org

Table of Contents

More Nuts and Bolts 22

Before getting into our discussions on topologies, protocols, media access, open systems, and fieldbus, it will be helpful again to brush up on some more nuts and bolts for those not generally exposed to this technology. Those of you who already work in this field can skip this chapter, unless you wish to be amused by my efforts at simple analogies. It is the area in which I get the most curious questions in class.

The Nature of Digital Communication

The concept of using a single wire to communicate data is not new. In fact it goes back to the use of the Morse code, which is really a form of digital communication. In it's early days they called it dots and dashes. Today, we refer to it as "ones and zeros," or two different energy states. Usually "one" is a higher energy level, and "zero" is lower energy level in this application. In the early days of telegraph, long-distance transmissions required a method for tolerating significant drops in energy levels. Even if the energy level (the sound) was weak, one could still determine when the signal was there and when it was not.

In the United States, one of the first uses of the telegraph was to communicate between railway stations along the nation's many hundreds (later thousands) of miles of track. A communication between the stationmasters at each stop along the line would report the status of the flow of traffic along that railroad. The communication between them was a series of short and long signals representing dots and dashes, called Morse code. Using dots and dashes enabled them to distinguish communication from no noise at all, that is, the users could distinguish between the two energy duration's—the dot and the dash—and a complete disconnect or broken wire.

An Example

The letter *C* in international Morse code is "dash-dot-dash-dot" and the letter *K* in Morse Code is "dash-dot-dash". If I were sending the same two in ASCII (American Standard Code for Information Interchange) code, however, I would send "1001011" for a C and "1000011" for a K. This is because in ASCII it takes seven bits to define a character. Other codes include the five-bit Baudot code, IBM's six-bit Correspondence code, and the eight-bit EBCDIC (Extended Binary Coded Decimal Interchange Code). These are just some of the many variations for defining characters, letters, numerals, and punctuation in different fields of endeavor. Of course, the many codes were each developed for different purposes. Their use would determine how large the space had to be to allow the use of more complex characters or combinations of characters.

Let's go back to our railroad communication again. Let us say the station-master in station 3 wants to tell the station-master in station 7 that the train is coming and that it is necessary to go out and prepare the water tower for refilling the locomotive tender with more water. In Morse code, he might send the message, "This is station 3 calling station 7." Everybody along the line would hear this message, but because he addressed station 7 only the stationmaster at station 7 would stop. He might put his lunch aside, lay down his newspaper, go to his telegrapher's key, and respond "This is station 7, what is your message, station 3?" He thus confirmed that he is connected, which in today's terminology is called "handshaking." Then station 3 would send the message "Go outside and check the level of the water in the tank; the train is coming." Station 7 would respond to Station 3, "You asked me to go out and check the water level in the tank; the train is coming". Station 3, having heard this full reply would know the correct message had gone through, and would give station 7 a message to begin the procedure. Today we call this an "echo check." Station 7 could also have said, "I heard your message of 18 words." Station 3, knowing that Station 7 had heard 18 words, and not 17 or 19, would know that statistically it was very likely that he received the correct message, and it was therefore safe to permit station 7 do the job. If he heard station 7 say some other number, then station 3 would resend the original message.

There are many kinds of checks and balances in sending communication, and they all mimic how humans convey information between each other. We humans continually acknowledge the fact that we receive a message from somebody by using some kind of a feedback signal. In the case of Morse code, the feedback has to be in terms of the Morse code technology. In the case of digital codes of any kind, whether it is ASCII or any other method, the feedback has to be in the terms of that code. This is known as "protocol."

For the protocol on this particular railroad, it might have been decided that there are many times for a station to call another station and ask them to check the water. Perhaps instead of keying in all those characters for all those words and sentences, it would be more efficient to reduce this lengthy string to only two characters. Everybody on that railroad would be taught that perhaps the characters CW would stand for "Check the water." This code stood for the entire procedure, yet communication would move much faster and there would still be no ambiguity. Many other "shorthand" codes would also be developed.

Sometime later, the stationmaster left his job and went to work for another railroad where they also had discovered it was easier to send just two letters. The two letters they picked might have been different however. Maybe they used the code LT for "Look at the tank." That would be considered a different protocol, but it would result in the same series of actions. The new stationmaster would have to learn this new protocol for all of the common signals that this new railroad normally sends.

Learning the protocol is what happens when you mix the communications of a Vendor A with the communications of a Vendor B. They're talking a different language. They may be using the same wire, they may be using the same telegraph keys, they may even be using the same (Morse) communication code, but the significance of that code may differ. There are several "layers" of communication involved, and these must be reflected within any type of digital communication technique that we use. In the following paragraphs, let's look for all the variations those differences can take.

Data Highways and Dirt Roads

Digital communications over a single wire, known as a data highway, have saved in excess of 30 percent of installation costs compared with hardwired connections for the same functions. A single consistent cable throughout a plant site, using one consistent connector, can be connected in only one way. It either is connected or it isn't. The only ordering and installation consideration is whether it's long enough. Installers need to know only "that box connects to this." There are no multiple wires to cross, and no surprises at start-up! The real cost is not in all the wire saved, which in itself can be significant. The savings is in all the cost-per-termination charges and the cost in all the checkout time required for each wire to "ring out" each line.

- Digital communications saves installation costs
  - Less wire, **fewer connections**
  - Less error in the wrong connections!
  - But … avoid narrow bridges, toll booths, and poor road surfaces and capacities.

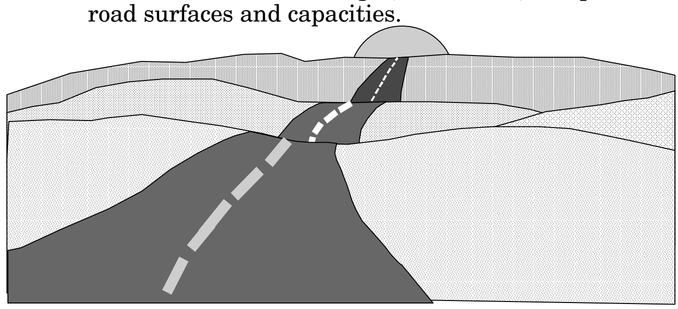


Figure 22-1. Digital Communications Differ From Analog; Advantages, However, Come with Caveats

Digital communication over a data highway can run at the "blinding speed" of 10 and 100 megabytes (MB) per second. This multilane superhighway is no use to you if the only way on or off is through narrow bridges, tollbooths, and poor road surfaces and capacities! Too many routers, bridges, and gateways can impede data flow and be a source of vulnerability. Many recent developments are improving this, but try to avoid sending critical messages through too many links between different networks and protocols. Quite often these networks and protocols must use "store-and-forward" techniques that cause serious delays, and they are not always able to be redundant. These devices will be discussed in more detail in Chapter 25.

Information Transmission

What is being transmitted when one device on a distributed control system (DCS) communicates over a "data highway" (communication cable) with another device? In the final analysis, energy is transmitted. The information exists in the form of binary numbers, a string of ones and zeros. That is the code for a number or a word or a condition. The dots and dashes in the code are actually two different energy durations. Generally, the ones and zeros that are communicated today are actually two different energy levels, such as amplitude differences or frequency differences. These ones and zeros are called “bits,” and they are usually stored in-groups of eight or multiples of eight. We now have 16 bit, and 32 bit and even 64 bit transmission.

Energy that at one level defined to represent the one and at another level to represent zero is sent over the communication medium. The form of the message may change several times before it arrives at its destination, but a message always consists of a block of time divided into segments, in each of which there exists a discrete energy level. From now on, we will refer to the transmission of bits but each bit is represented by an energy state. Since time is a factor, there will always be some sort of clock keeping tabs on the message transmission.

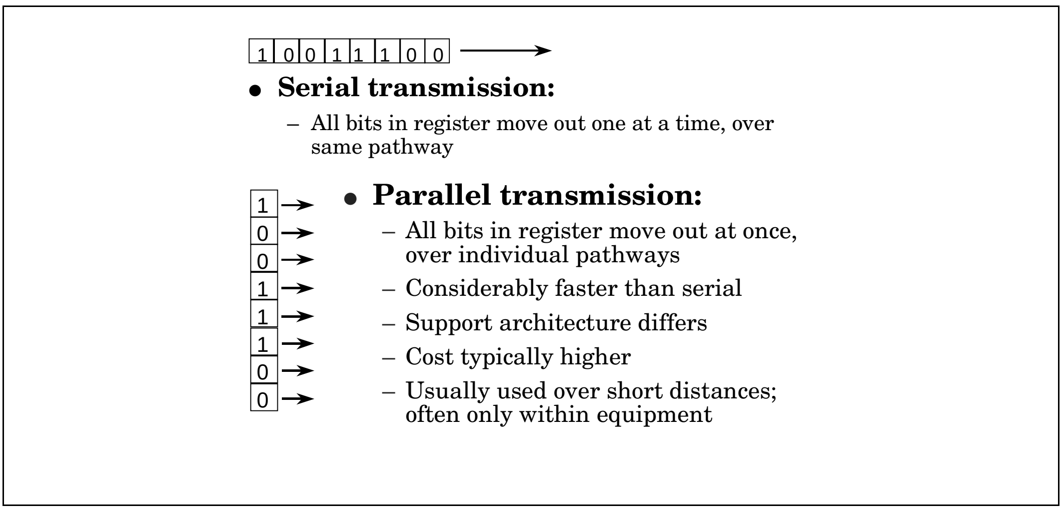


Figure 22-2. Parallel or Serial Communication: Fast or Far

Parallel Transmission—A transmission mode that sends a number of bits simultaneously over separate lines, such as sixteen bits over sixteen lines; usually unidirectional.

Serial Transmission—The most common transmission mode, in which information bits are sent sequentially on a single data channel (wire, radio beam, etc.).

Parallel versus Serial

If the entire sequence of bits can be sent in parallel, the transmission can be very fast. Parallel transmission is often used on printed circuit cards when an entire word is sent over a bus made up of eight (or some multiple of eight) parallel sections. These are usually foils on the printed circuit board, or on the backplane of a controller card cage or other device. Over short distances, ribbon cables, which are flat cables with copper conductors running side by side and insulated from each other with a multipin connector at each end, can carry the signals corresponding to the bits making up a word, in parallel, between devices. For the short distances on a printed circuit card, where so much is happening that time is measured in nanoseconds, parallel transmission is practical and necessary.

Transmission over distances between devices more than a few feet apart becomes prohibitively expensive if a separate wire is used for each bit. Also, there is some loss in energy level as an electrical signal overcomes the resistance of a wire or the conductor, and after a certain distance signals lose strength and must be amplified. The complications involved in amplifying a number of parallel signals and maintaining the relationship between them are great. That is to say, if each of the bits placed into the wire on one side goes a long enough distance, they may not all come out at the other side at the same time. This would impact the significance of that message.

> The complications involved in amplifying a number of parallel signals and maintaining the relationship between them are great. That is to say, if all of the bits are placed into their respective wires on one end of a very long cable of wires the naturally varying individual resistances could easily cause them to reach the other end at different times! This would significantly impact the meaning of the message that is transmitted.

Consequently, most transmission between devices is serial. This means the bits in a register are moved out one at a time sequentially sent on their way, then put back into a register at the other end. For a serial electrical transmission, only two wires are required, one to carry the energy and one to act as a common signal ground. In distributed process control systems (DCS), this has been called the data highway.

Interference with Transmission

Electrical transmission is always subject to the effects of disturbances caused by electrical energy on the wires or near the wires that is not part of the signal being transmitted. In transmission cabling, three types of interference are often encountered. They are magnetic interference, electrostatic interference, and crosstalk.

If a conductor carrying current is near another current-carrying conductor and the current flow in the second conductor changes significantly, there will be a corresponding voltage change in the first conductor. The characteristics of the signal will change because there has been an energy level change. This type of interference is called magnetic interference, because the change in current flow creates a varying magnetic field around the wire. Cables and wire bundles carrying electricity at different power levels or different frequencies should be separated far enough apart to prevent this kind of interference. Some standards call for an eight-inch separation.

An electrical charge resulting from current flow into an inductive device, such as a relay or solenoid coil, or from a welding machine can produce an electrostatic electrical charge on a wire. This introduces spurious signals and can disrupt transmission. Surrounding the wire, or wires, in a cable with conductive shield that can be grounded is one method for reducing this type of interference.

When alternating current (AC) signals or pulsating direct current (DC) signals are transmitted on a pair of multipair cables, it is possible for the signals to be superimposed on other signals carried by adjacent pairs. This is called crosstalk. To help prevent this, each pair in a cable may be twisted and shielded. The twisting will cancel out the unwanted signals, and the shield will carry them to ground.

Another type of problem arises when different points of the circuit are of different ground potentials. When this happens, currents can be circulated from one part of the circuit to another, creating voltages that may be read as significant signals. To eliminate this problem, suppliers require one, and only one, common ground for the electrical system. This is an important requirement and particular attention should be paid to it when equipment from more than one supplier are interconnected in a system. Do not overlook, however, checking equipment from the same supplier as well!

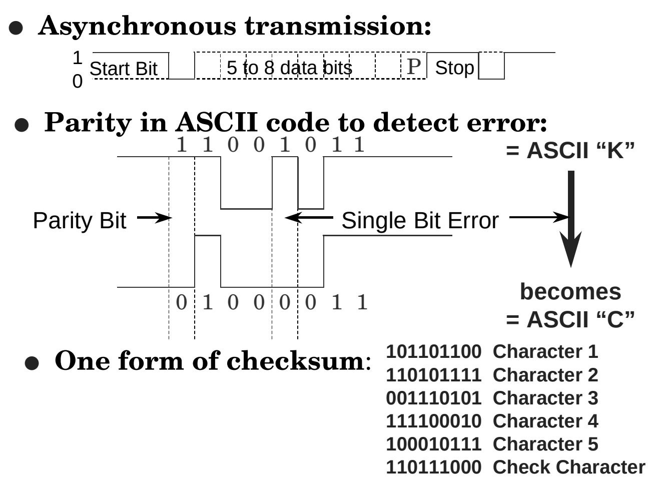


Figure 22-3. Confirming Good Data Transmission

---

asynchronous—Computer logic or communications in which all operations are triggered by a free-running signal not related to specific frequency or timing. Successive stages are triggered by the completion of the preceding stage.

ASCII—[asky] American Standard Code for Information Interchange: Binary character code, each representing a single computer character, to define 128 uppercase and lowercase characters, numerals, punctuation, and special communication control characters. A standard method of encoding characters into seven or eight binary bits, typically 7-bit-plus-parity code, representing alphanumeric data for processing and communication compatibility among various devices. Defined in ANSI X3.4-1986 and normally for asynchronous ransmission.

checksum—The summation of digits or bits according to an arbitrary set of rules used primarily for checking the integrity of data; can detect single bit errors and some multiple bit errors.

parity bit—A bit that is set at "0" or "1" in character to ensure that the total number of 1 bits in the data field is even or odd in digital communication between devices or components.

parity check—The addition of information bits that make up a transmission block to ensure that the number of 1s is always either even (even parity) or odd (odd parity); used to detect digital transmission errors.

Physical Structures 23

Connecting individual devices in the most expedient way is the goal of any network technology. The network should be considered in terms of speed, directness, cost, simplicity, and reliability. Generally, in industrial control systems three popular topologies have been successfully used. They are the star, the bus, and the ring network systems. There are several other network systems used in other industries, but we will not dwell on them, as they are not very common in the current industrial control scene.

In the control of a single industrial process unit, or in a laboratory setting, a network is a system that links a few controllers and perhaps a common operator interface. Different communication requirements are needed to network a plantwide system. There are even different challenges for a system that must reach beyond the plant for long distances, such as you would find in water treatment, power distribution, or gas distribution lines.

Network Topologies

Since the late 1980s, the single loop control networks have been emerging. They, of course, would be the simplest of systems because they were only intended to be used in the laboratory or on an isolated unit process control requirement. In recent times, however, the single loop controllers have been available with multiple loops. More sophisticated control capability is expected on these small systems. They are no longer merely linking a handful of controllers perhaps to a small personal computer (PC) used to manage some reports. In the mid 1990s, this PC has gone beyond report writing to become a local operator station and is even used for configuring other control strategies.

More typical are the network systems associated with traditional distributed control systems (DCSs.) These have generally been designed for large multiunit process control and plantwide control systems. Most of the current topologies have been designed for this scale and grew from systems driven from single mainframe computers. The design of these has had the greatest need for redundancy.

On a larger scale, "traditional" supervisory control and data acquisition (SCADA) systems have been used for control actions and information gathering from *beyond* the plant. For over four decades these "traditional" SCADA systems have not been used in process control but rather in the starting and stopping of remote units, such as those found in remote power transformers or in remote water or gas pumps on pipelines. The communication, quite often, is not through wire, but through radio transmission, phone lines, and even satellites. The time delays on these SCADA systems have usually made it necessary to rely not on monitoring and controlling the details of the process itself from a distance. The "supervisory control" portion of these systems was only expected to merely turn specific units on or off or to bypass units that may have been damaged, for instance, in a storm or in an accident. Any communications of such remote transmissions expected by a SCADA system would have to allow for long time delays between the request for action and the action to happen. This generally precludes any continuous process action, which requires a more responsive operation. As was stated before, do not confuse these comments on SCADA with those on data acquisition and control (DAC) systems used *within* a plant, which some vendors have recently been calling "SCADA."

bus—Transmission path or channel; an electrical connection with one or more connectors through which all attached devices receive all transmissions at the same time. In a linear network topology such as Ethernet or token bus, all network nodes "listen" to all transmissions, but each responds only to those addressed to it.

multidrop—A network topology of computing devices or multiple field devices that are connected to one pair of wires, or a coaxial cable, usually attached with "tee-connectors."

ring network—Network topology in which each node is connected to two adjacent nodes, the entire network forming a closed ring; communication between any two points must include the intermediate points.

star—Network topology of computing devices in which each station is connected by communication links radiating out of a central hub that handles all communications.

topology—Logical and/or physical arrangement of stations on a network, such as star, ring, multidrop, tree.

These larger systems, distributed control and SCADA, generally may employ several topology types within the same network. They're generally considered a plant network that is connected by several subnetworks, and these in turn can be connected to other subnetworks. Their structures would depend on the requirements of the environment in which they are to operate. For example, communication between a control room and the rack room, if they are only feet apart as was traditionally the case in older plants with single mainframe computers, would not involve the same requirements as would the more common practice today of having the central control room communicate to distant control stations located in separate rack rooms.

Furthermore, each of the control stations can have a network of input-output (I/O) terminals. Initially, the (I/Os) were analog up through to the controller. Today, with the many smart sensors, transmitters, and final elements, these networks are frequently digital.

Additional links from the control room to a plantwide network of business management computers usually require another network. These do not need the speed of process loops but rather the capacity of a large information-handling system.

As a result, in this plantwide system there are many small local I/Os with relatively low volume of data in each, but response times must be in microseconds. On the other end of the same system, the business management side of the plant requires large volumes of data, but the response time can be in hours or even days or beyond. As a result, the topology, the immediate access technique, and the protocols used will differ on each portion of this system (within each subsystem).

Star Topology

This method, shown in Figure 23-1, is the simplest and therefore the least expensive of the topologies we will be discussing here. In this technique, a single active switching device can be viewed as the center of the system, and it has a link to each of the control system components. This control system component can be an active loop for a discrete control station and/or it can be an operator station. It could be that each of those controllers has an operator interface incorporated in it or it could be that one of the devices is a centralized operator station for managing each of the other controllers. In the latter, quite frequently the operator station is incorporated with the central active switching device.

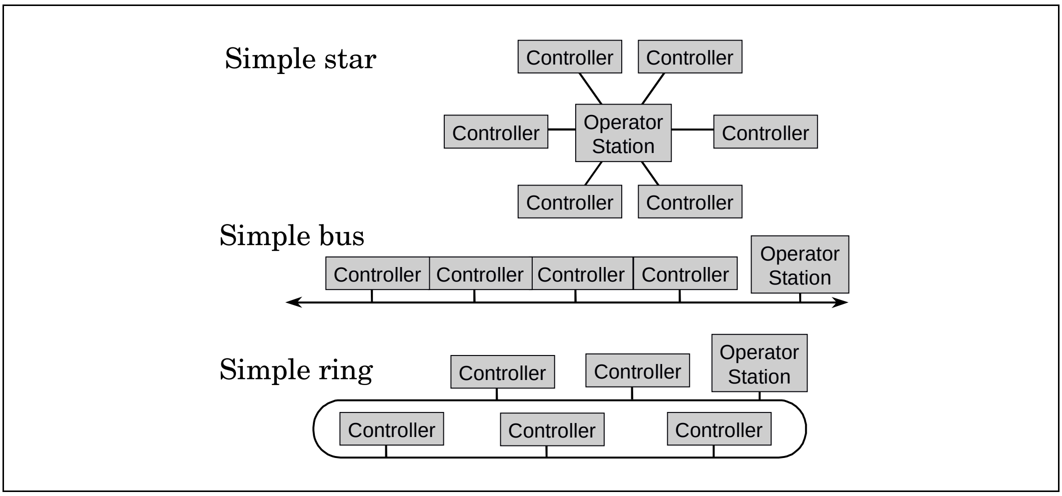


Figure 23-1. Many Control Systems Have Combinations of These Network Topologies

Usually, these systems are not redundant and redundancy starts to increase their price. They are often used so that each of the control elements can operate independently and are not required to communicate all the time. In this way, they can tolerate a loss of communication for a short or even a prolonged period of time. Such a system is probably more suitable for a small laboratory or a single isolated unit process, which normally would not be connected to the rest of the plant. Of course, each of these systems can also be a subnetwork of a larger system. In this case, the larger system would be connected through that central active switching device.

In an attempt to overcome some of the disadvantages of the star topology, a mesh topology has been developed in which each of these devices on the network has its own switching device within and provides communication between every device one to one. This causes a crosshatch of communication, with each box having a single link to each of the other boxes, giving the “mesh” appearance but, of course, increasing the complexity and naturally the cost. The mesh improves the security of the system, but it certainly is not the most expedient way to do this in today’s technology, so it is rarely seen in industrial process control.

Bus Topology

In this topology, shown in Figure 23-1, all the messages pass through a common piece of hardware like the star topology, but this common piece of hardware is passive, generally a single wire or a communication cable. The active switching occurs in each of the elements on the bus. There are versions of this which there is a switching device called a traffic director as a separate device on this system, and we'll discuss that when we discuss media access protocols in Chapter 24. As with the star topology, there is a vulnerability should the bus fail. There are methods in this system, however, for a redundant bus that do not involve that cost disadvantage of the redundancy put into the star. Sometimes this bus topology has been called multidrop systems.

Ring or Loop Topology

In this approach, shown in Figure 23-1, a communication cable is linked to each of the control devices on the network, one after another. Sometimes they are daisy chained, meaning that each can only talk to its adjacent station, and other times other techniques of communication allow each to talk to any of the other boxes. Protocols and media access techniques provide the difference.

The simplest arrangement is to have the communication move around the loop in a single direction. Quite often, in the redundant version, an active second communication ring connects each of these devices so that communication is moving in the opposite direction simultaneously with the motion of the first cable. This is not an expensive technique for redundancy and was becoming more and more popular in the systems of the 1990s.

Network Cable

Local and global networks for process control systems are made of electrical wire cable or a fiber optic cable. Electrical wire cables commonly used in the industrial environment include twisted pairs, coaxial, and twinaxial cable (Figure 23-2).

UTP—Unshielded twisted pair: Wiring for signals consisting of at least two conductors twisted together six twists per inch to minimize the effects of electromagnetic radiation between them but without a metal covering to protect it from external EMI or RFI.

STP—Shielded twisted pair: Wiring for signals with at least two conductors twisted together six twists per inch to minimize the effects of electromagnetic radiation between them and covered with metal-backed Mylar, plastic, PVC, or metal-woven sleeve for protection from external EMI and RFI.

coax—Coaxial cable: Popular transmission medium that is formed from two or more coaxial cylindrical conductors insulated from each other. The outermost conductor is usually grounded and encased in either wire mesh or extruded metal sheathing. It is frequently used for television and radio signals as well as digital signals because its design is less likely to cause or be affected by external fields. Many varieties are available depending upon the shielding needed and the voltages/frequencies to be accommodated.

twinax—Twinaxial cable: Uses a twisted pair of conductors within shielding to improve resistance to RFI/EMI over coaxial cable; see coax.

fiber optics—Transmission technology in which modulated light wave signals, generated by laser or LED, are propagated along a (typically) glass or plastic medium, then demodulated to electrical signals by a light-sensitive receiver.

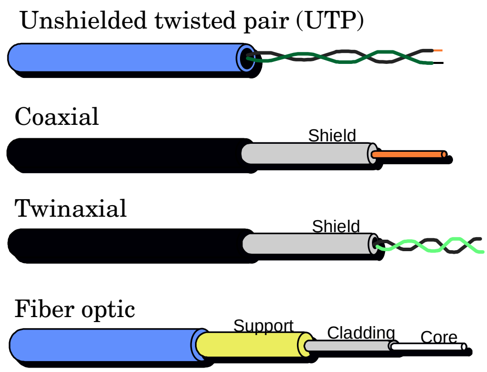


Figure 23-2. Most Popular Cable Types Used in Control Systems

As described earlier, twisted pairs reduce cross-talk and each pair is usually shielded so that magnetic and electrostatic interference is reduced. A shield may be a woven metallic wire or a metallized polyester material. It is usually combined with a drain wire that can be terminated at a ground connection. A number of twisted pairs can be combined into a cable, and it is a good idea to include extra pairs in the cable to accommodate circuits that may have to be added after installation is complete. Twisted pairs are less expensive than the other types of network cable materials, but their range is also less because of attenuation as well as capacitance effects.

Coaxial cable must be handled carefully when it is installed. If the outer tube is bent or crimped, the cable can be changed, degrading the signal.

Coaxial cable has a central conductor surrounded by an outer tube like conductor. The outer conductor, or shield, is used as the common signal ground. It has distinct advantages over twisted-pair conductors when high frequencies and high bandwidth are characteristics of the transmission system. Bandwidth is a function of the number of signals that can be handled by a network in a given period of time. The loading factor of coaxial cable is the attenuation due to conductor loss and dielectric loss. Coaxial cable must be handled carefully when it's installed; if the outer tube is bent or crimped the impedance of the cable can be changed, degrading the signal. Properly installed, it has a long useful life and can carry large amounts of information over long distances. A variation of coaxial cable called Twinax provides two coaxial cables in a single jacket. Twinaxial cable provides more shielding from noise than coaxial cable and is more expensive.

Coaxial cable does a very acceptable job as a communication network media. It is more expensive than twisted pairs but less expensive than a glass cable. Why do fiber optics merit consideration? Significant is the fact that the optical fiber cable does not carry electricity and eliminates all the problems inherent in interference with electrical transmission (Figures 23-3 and 23-4). There are no ground loops or common load voltage problems. Electrical noise is not a problem. There is no inductive pickup, cross-talk or interference from transients. It will not “short out” in puddles and is far more resistant to chemical corrosion than any metal carrier. The cable is lighter and easier to handle than electrical cable, making it far easier to hang on poles and to pull through wire trays and other passages. Under most conditions, there is little possibility of a spark in explosive and inflammable environments (Protection in hazardous areas is governed by energy levels. The energy needed to operate control systems is rarely as high as that needed for interstate phone systems.)

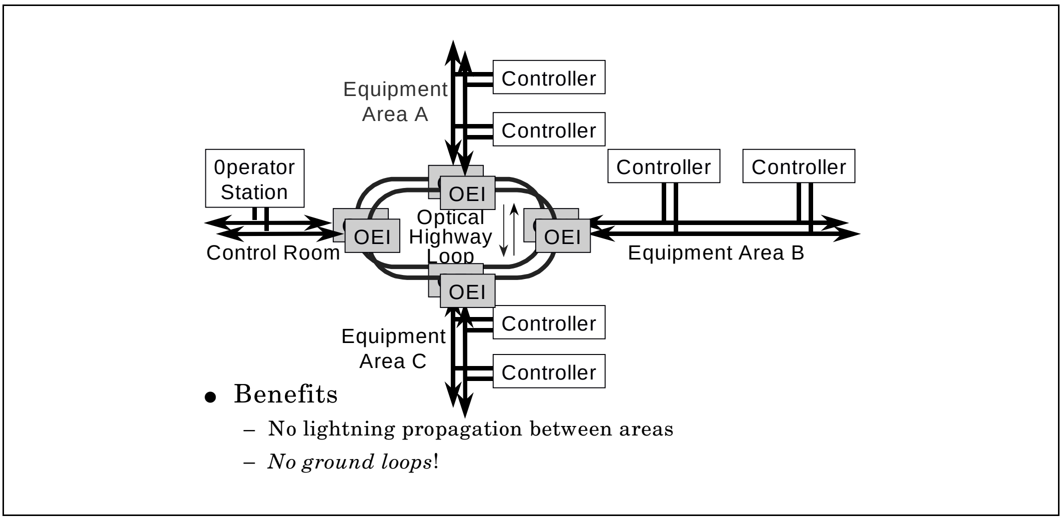


Figure 23-3. Non-Electrical Network Provides Signal Protection

Interestingly enough, because there is no electricity fiber-optic cable is not subject to electrical codes nor must it be installed by electrical crafts. The skill level for installation is very much the same as for coaxial cable. Today, splicing and terminations are done just as easily. Phone companies have laid fiber optic around the world using local labor.

More on Fiber Optics

Fiber-optic communications have been used in control systems since the early 1980s. Glass fibers instead of copper or aluminum wires are the conductor, but energy is still what is being conducted. Instead of electrical energy, light is transmitted in pulses. These pulses can be at different frequencies, and this produces a bit of information. A high frequency, a low frequency, and a neutral frequency (carrier wave) are used to distinguish the bits. Rather than detect absolute voltage levels, the receiving circuit needs only to detect the number of zero crossings in a given time to determine the state of the transmitted bit. Light energy travels at the same speed as electrical energy in this

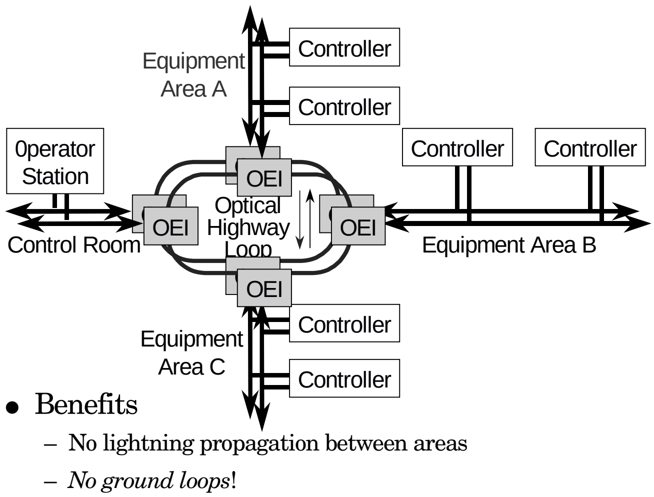


Figure 23-4. Used *Between* Cabinets, Each of Which Has Protection For Their Contents

condition and can represent binary data equally well. (Speed of light is far slower in a medium like glass than in a vacuum, and the more significant limitation in fiber optics is the electronics at either end of the length.)

Once a pulse of light enters a glass fiber, it cannot leave until it gets to the other end. When light travels from one material to another, it bends. A straight rod stuck into clear water appears to bend below the surface of the water because the light carrying the image passes from liquid to air. The bending is a function of a measure called refractive index (Figure 23-5). If the light travels from a material with a high refractive index to one with a low refractive index, it will not pass through the interface, but bounce back, reflected from the surface back into the original material. By making glass fibers with a high-refractive-index inner core, and a low-refractive-index outer sheath, light can effectively be confined in a rod, escaping only at the end. There are many medical applications for this phenomenon.

---

fiber loss—Attenuation (deterioration) of light signal in optical fiber transmission.

graded index fiber—Optical fiber whose core has a non-uniform index of refraction; the core is composed of concentric rings of glass whose refractive indices decrease from the center axis to reduce modal dispersion and thereby increase fiber bandwidth.

refractive index—Ratio of the phase velocity of light in a vacuum to that in a specified medium.

optical fiber—Any filament or fiber made of dielectric materials and consisting of a core to carry the light signal and surrounding cladding that reflects the signal back into the core. A thin glass thread is most commonly used, but plastic fiber can also be chosen.

bend loss—A form of increased attenuation in optical fiber that results from bending fiber around restrictive curvature (microbend) or from minute distortions in that fi**ber (microbends).**

bend radius—Smallest arc in cable that can be made without causing damage; fiber-optic cable is no worse than coax. In fact, it is often superior, depending upon the sheath, not glass or cladding.

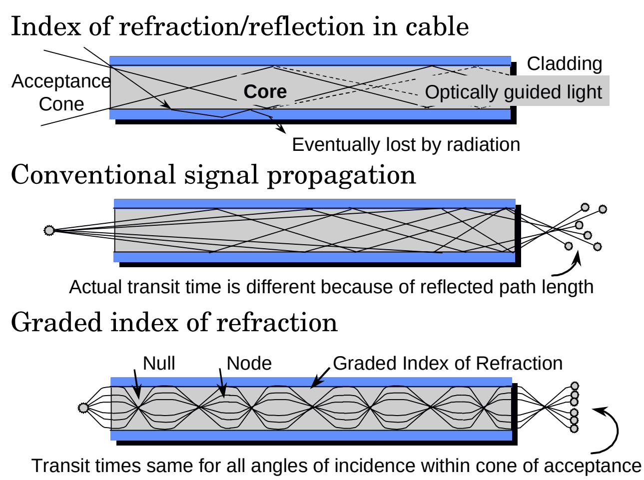


Figure 23-5. Refraction/Reflection Are A Consideration in Fiber Optic Cable, Not Voltage Drops and Current Loss

In the 1950s, when this fiber-optic application was developed, glass was not pure enough for communication transmission. Small imperfections weakened and blurred the signals, so they could travel only short distances before becoming unintelligible. Transmission losses were a thousand decibels per kilometer. Bell Labs and Corning Glass worked on the problem and almost simultaneously developed a process in the 1970s that today produces glass fibers with purity so great that losses are listed as less than ten decibels per kilometer. In the drawing process developed by Bell, a three-foot rod of glass production stock ends up as nine miles of fiber, five-thousands of an inch in diameter. A typical fiber-optic cable suitable for distributed control system network has a diameter of 75 to 100 microns.

The speed of light is far slower in a medium like glass than in a vacuum. The more significant limitation is that of the electronics at either end of the length.

Light travels directly down the center of a fiber-optic cable will emerge faster than light that bounces off the sides. This will cause the signal to break up over a distance as the cable flexes. To overcome this phenomenon, glass layers are deposited around the drawn glass core as it is made. The effect, called multimode (refraction), calls for the resulting cross section to act as a lens that redirects the light toward the center as it migrates outward. Most of the light signal stays intact making longer lengths of cable practical.

At the terminations, an optical/electrical interface (OEI) device will send electrical pulses to a LED, which then converts that electrical signal to optical pulses over the optical cable. On the other side of the cable a PIN diode, which is a light-sensitive device that develops an electrical signal when exposed to light, will take those optical pulses and convert them back to electrical. Of course, there is some circuitry to bring the energy level to the same electrical level as the device requires. These operations are merely energy conversions; they are not a store-and-forward of information and cause no appreciable delay in the signal. The same optical/electrical device can function as an amplifier/relay to extend the cable lengths.

Actually, three different signals are involved in the particular system shown in Figure 23-6.

- The message itself (communication signal); two-directional both send and receive
- The one that disconnects the ring (solid-state switch) during transmission so the signal doesn't go around "forever"
- The one that listens for the signal to return to assure the entire ring is still connected

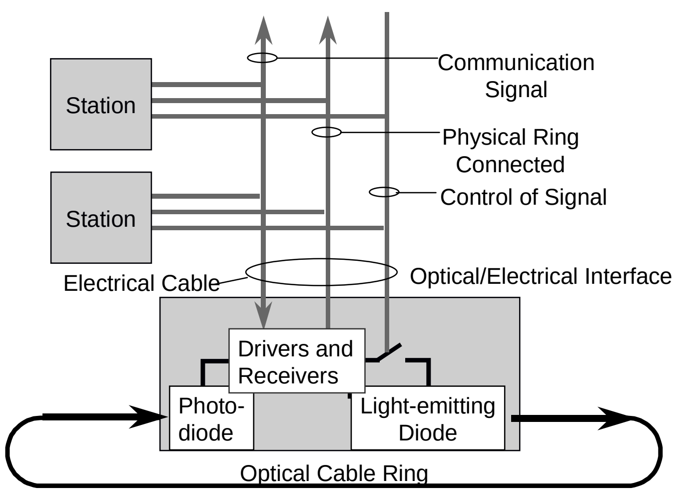


Figure 23-6. One Method of Linking Optical Medium with Electrical

Logical Structures 24

In the early days of digital communications, there was no question which device was the transmitter and which was the receiver. A computer sent data to a printer, for example. Usually, a single cable connected them, and transmission was simplex, that is, it went in one direction only. In a distributed control system (DCS) today, information must move between operator stations and remote controllers in both directions. This would require two wires in a simplex network. In single-wire systems, communication would occur either as full duplex, a complex technique by which both devices talk at the same time, or half duplex, where each takes turns sending and receiving, very much like a question-and-answer session between people (Figure 24-1).

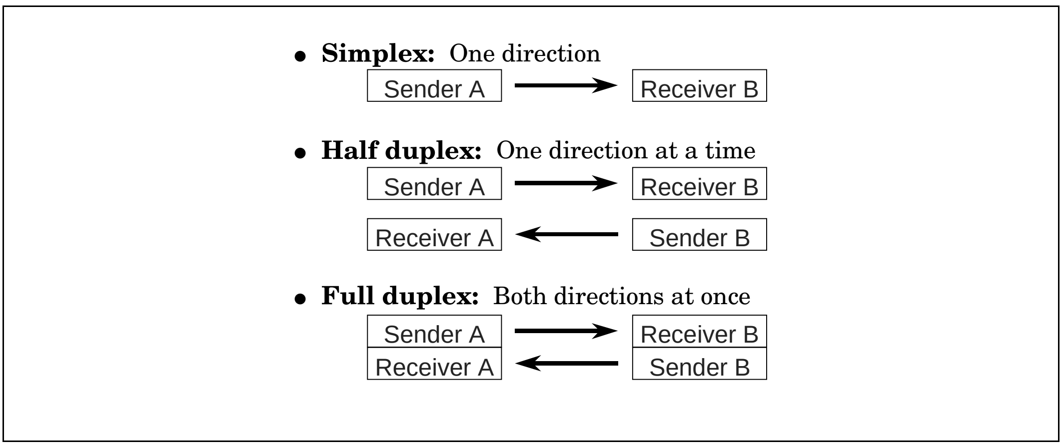


Figure 24-1. People Also Talk This Way: Lecture (Simplex), One-on-One (Half Duplex), and at a Party (Full Duplex)

---

simplex cable—Term sometimes used for a single fiber-optical cable.

simplex mode—The operation of a channel in one direction only with no capability of reversing.

simplex transmission—Transmission in one direction only.

duplex cable—In fiber optics, a two-fiber cable suitable for duplex transmission. With copper wire, a pair of wires insulated from each other and with an outer jacket of insulation around the inner insulated pair.

duplex transmission—Transmission in both directions, either one direction at a time (half duplex) or both directions simultaneously (full duplex).

Message Format

In asynchronous transmission, the sender and receiver use a count of bits to define the transmission interval. A code often used to transmit characters is ASCII. This 7 bit code provides numeric characters, alphabetic characters, uppercase, lowercase, punctuation, and special characters such as carriage return and line feed. These last two are a carryover from the time this code was used with Teletype. (Recall our earlier comment about the ASCII characters for the letters C and K comparable to our Morse code.)

Block-oriented synchronous serial transmission can be more efficient than asynchronous and consequently is used for almost all network use. The transmitter sends an entire block of information at a time, preceding it with specific groups of characters and ending it with another specific group of characters. By using synchronous transmission protocols, efficiency of transmission can be increased as much as 95 percent. On the other hand, if blocks contain only a few characters, this advantage is lost. We will discuss a little later how we can improve the speed of transmission, but for now, let's take a look at how to use synchronous transmission in an efficient way.

The specific group of characters at the front of this block is called a header and contains a synchronizing character as well as an address and sequencing information. The synchronizing character is a recognition device. It acts as a punctuation mark in a line of script, or like the serial number on a $20 bill. The receiver searches for the synchronizing character and, as soon as it appears, uses it to lock into phase with the characters that follow. A group of characters following the message, or text field, contain error-checking instructions and an end-of-message character.

Synchronization is established and maintained by a dialogue between sender and receiver, called "handshaking." Special characters are sent back from the receiver so that the sender knows that his or her message was received or was not received intact and should be repeated. This uses up transmission time, of course, but the loss is relatively small compared to the time saved by sending a large block of data.

In binary information there can be any combination of bits, and a random 8 bit piece of data could have exactly the same bit pattern as that of a control character. It would not do to have the bit pattern correspond to the control character for the termination code "end of message" to show up as a legitimate text character in the middle of a transmission thus causing the receiver to terminate receipt. Each protocol must have a way of transmitting text that may contain bit patterns that look like control characters to the receiver. This is called "transparency." The protocol may call for a control character to be always preceded by another character. In some protocols, this is called a data link escape (DLE) character. Was STX (for "start of transmission") preceded by DLE? If it was not, then it is text. The practice carries over into the data stream itself. Here is a DLE character. Was the preceding character another DLE? If it was, it's legitimate for text. If not, the following character is a control character. Fortunately, the chips have the software for transparency built into them, so all this happens automatically.

A number of synchronous protocols are used by distributed control systems (DCS) suppliers and by computer manufacturers. Commonly used are the following:

BISYNC—binary synchronous communication protocol; developed by IBM.

DDCMP—(Digital Data Communication Message Protocol), which uses a full duplex form of transmission and is compatible with the DEC PDP/11 series of minicomputers.

SDLC—(Synchronous Data Link Control) [Figure 24-2]; bit-oriented standard protocol developed by IBM superseding bisynchronous transmission; uniform discipline for transfer of data between stations in point-to-point, multipoint, or loop arrangement, using synchronous data transmission techniques.

HDLC—(High-level Data Link Control) [Figure 24-4]; international standard communication bit-oriented protocol defined by CCITT for ISO, and used in Open Systems Interconnection (OSI).

ADCCP—(Advanced Data Communication Control Procedures) developed by ANSI (American National Standards Institute).

Some form of SDLC or HDLC is most commonly used in suppliers' specifications.

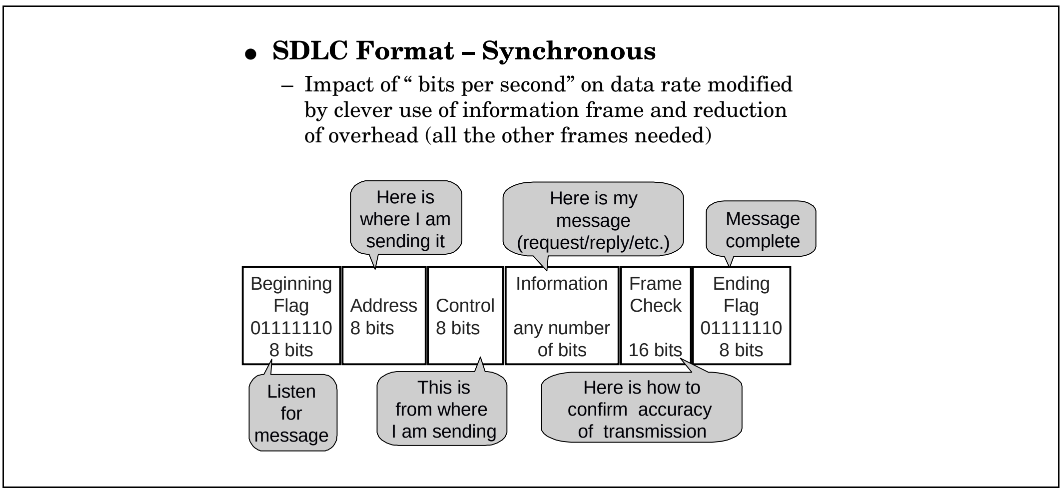


Figure 24-2. Typical Transmission Example

The creative use of scan groups by vendors in proprietary networks helped them achieve the "real" real-time so essential to process control use (Figure 24-3). In this approach, rather than using the same overhead for each parameter, all the needed parameters for the screen view (or other purposes) are clustered into useful scan groups, each group sharing the same overhead. A single scan group could have eighty items, saving eighty times the overhead on a single transmission!

For example, an operator could push a button for information on a screen view. This could result in a different scan group, depending upon what view is being invoked:

- Overview needs only deviation values and alarms for display
- Group view additionally needs "faceplate" data (PV, SP, deviation, alarms, etc.)
- Detail or Point views need to add all the tuning parameters and other settings

- Typical "scan group" could be all needed parameters of single control loop in one specific controller on network:
  - Amount of information determines scan group size
  - Assumes 1 byte = 8 bits; Assumes 1 bit takes 2 µsec.

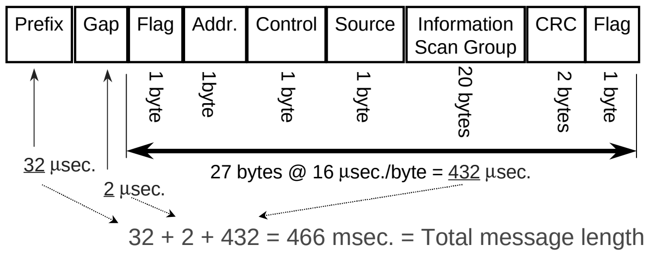


Figure 24-3. Use of Scan Group Can Help Maintain "Real-Time"

For this to work, however, the controller would have to be designed to match the operator station so that both recognize each of the scan groups. This can only work in a proprietary system. This could change some day if some very specific standards can "reach" into every system design. This kind of capability may be a future province of Fieldbus Foundation, which is the only fieldbus design of the 1990s that addresses the "user layer" of communication.

Media Access Protocols

In most process control networks, there are several stations sharing a common communication media, many of which need prompt access. It is imperative to arbitrate how the stations will communicate, the sequence they will communicate in, and which takes priority. Some of the accessing methods used by these protocols will only work with certain network topologies, for example, a ring topology or a bus topology. The purpose of the media access protocol is to control the access of each station to the shared communication media and to structure the format of the messages that go on this network.

When a cable system is shared, a protocol is needed to determine which station is granted access to the medium. If two stations transmit at the same time there is a collision. A protocol is required to minimize collisions.

The media access protocols that are typically used in distributed control systems (DCSs) are as follows:

- **Carrier insertion.** A method whereby a station in the network monitors a message stream of all the messages passing through it until it detects a lull in traffic. At that point it inserts its own message while buffering, and later retransmitting, any additional incoming messages. This is also known as *ring expansion* because the method expands the ring of data by one message until the original message, or the acknowledgment of it by the receiving station, returns back to the sender.

- **Carrier Sense Multiple Access with Collision Detection (CSMA/CD).** A method in which the contention between two or more stations is resolved by detecting any simultaneous transmission and causing each to retry after waiting a predetermined time. The time-out delay is different for each station on the network and is usually several microseconds. This is the method used by Ethernet, which is standard like IEEE 802.3- that is used by Intel, Xerox and Digital Equipment Corporation.
- **Polling.** A method of controlling devices on a bus or a ring (loop) communication network with a single network master, sometimes called *traffic director*. It is the process of inviting another station (node) to transmit data as compared to selecting that station as in TDMA.
- **Time Division Multiplex Access (TDMA).** A method by which a single device or multiplexer accepts multiple channels on a single transmission line. To do this, it connects stations one at a time at regular intervals, interleaving bits (called bit TDM) or characters (called character TDM) from each station. This multiplexer also acts as a traffic director, except that it arbitrarily signals (selects) each device, telling it when to talk.
- **Token passing.** A method for peer-to-peer communication. The right to transmit is passed from device to device, called a token, ideally on a regular schedule, making a throughput and response time predictable (deterministic) so that only one station has the token at any one time. During the time the station has the token, it has complete control of the network. Two versions of this are as follows:

1. The token bus (defined by IEEE 802.4), where the right to transmit is passed from device to device by way of a logical ring on a physical bus configuration, and
2. The Token Ring (defined by IEEE 802.5), where a special data packet is passed from station to station around a physical ring in sequence. When a station wants to transmit, it takes possession of the token, transmits the data, then frees the token after that data has made a complete circuit of the electrical ring.

Common access methods include Traffic Director, Token-passing and CSMA/CD. Traffic director and token-passing are deterministic, CSMA/CD is non-deterministic. If the network access time for a station can be calculated, the network is called deterministic.

Media Access Analogies

Some analogies of media access methods (Figure 24-4) can be: military (traffic director), a press conference or barbershop (for CSMA/CD), and Robert's Rules of Order at a meeting (for token bus) [versus Pony express rider carrying message packets from station to station (for token ring)].

In the military analogy, the traffic director operates for command headquarters when information is requested from the units. "Fall in and report," commands the leader (traffic director), who then polls who is (which controllers are) available for duty. Requested information from them is then relayed up to command headquarters

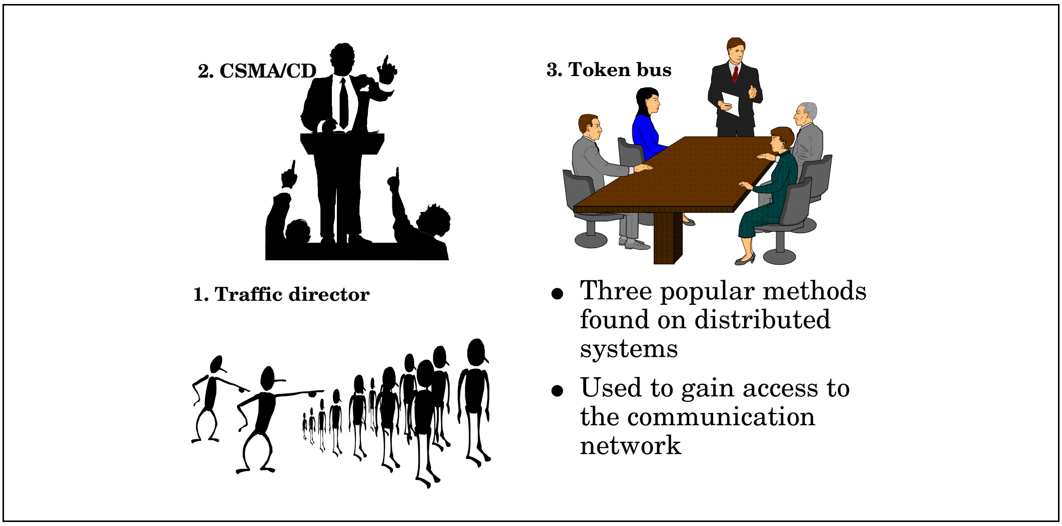


Figure 24-4. Typical Access Methods in Control Systems

(operator station). There is no sharing of data between the troops (no peer-to-peer communication between controllers), but everything must go up the chain of command, where it is then decided what data to disperse and where.

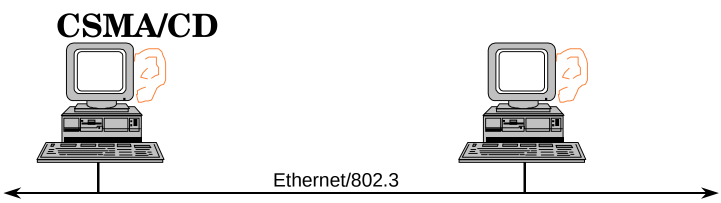
In the press conference analogy, the press corps (controllers) all frantically vie for the speaker's (operator station's) attention. When the speaker selects, only one of the reporters holds a dialogue with the speaker. As soon as there is a pause, everyone again retries for attention. The more prestigious reporters (who have the shorter time-outs) are higher on the list for access to the speaker. In a control system, the critical alarms must be allocated there.

In a token ring system, when a keystroke occurs at an operator station requesting information from various controllers, the Pony Express rider is sent out on a circuit of stations (controllers) with the message in his saddlebag. At each station on the circuit, he must stop, open the pouch, inspected to see if the message is for any activity at that station, add any new mail, close the pouch, and send the rider off to the next stop. This continues around the loop. If there are many stations and/or much more mail (many alarms, for example), then the rider can be slowed or even overburdened. He might end up pulling a wagon of mail, but with the same pony! Sometimes the pony does not live through the ordeal.

The token bus system, however, operates like a business meeting. Perhaps both the president and a vice president are present, along with the employees (controllers). Each has a turn to speak and/or ask questions, but when they "have the floor" they are in full control of the meeting. They could ask for a set point from one, a calculation from another, and a report on any alarms. The others can respond immediately when addressed; they do not have to wait until they have the floor (the transaction is completed right away). The sequence may be by the order of seating (physical ring) or by alphabetical order (logical ring). The managers (operator stations) each have an attendance sheet and make records as each employee takes his or her turn. When each is finished, they "yield the floor" (pass the token) to the next in turn (using a "handshaking routine). If there is no acceptance after a few tries (typically three), the previous speaker will pass over that person and go to the next in order, while the managers mark their attendance sheets (report some diagnostic error code).

Ethernet

CoDeveloped by DEC, Intel, and Xerox in 1973, the Institute of Electrical and Electronics Engineers (IEEE) created Ethernet as their 802.3 standard in 1983 (Figure 24-5). It only handles OSI layers 1 (physical) and 2 (data link). The other five layers are handled with other software.



Carrier Sense (CS) – Device listens for clear channel before transmitting. If channel in use (a carrier is sensed), device will delay transmission.

Multiple Access (MA) – Many devices can connect to common cable and have equal access when channel is clear.

Collision Detection (CD) – Two devices can sense clear channel at same time and begin transmitting. Collision will occur and both devices will stop transmitting, wait some random time different from each other, and each will try transmission again.

Figure 24-5. Ethernet is Claimed by Some to Be the Most Widely Used Network in the World

CSMA/CD: Every device has access (Multiple Access), with no traffic director required. A device that wants time on Ethernet first listens (Carrier Sense) until the highway is available. The device then tries to send its data. If another device simultaneously tries to send data, a collision occurs and both devices stop (Collision Detection). They will each retry after different delay times.

Token Passing

One station at a time has the right to transmit (Figure 24-6). That station posses the token. When the token-holding station is through transmitting is sends the token to the next station in the logical ring (Imagine a group of people standing in a circle but taking turns speaking in alphabetical order.) The time it takes for the token to be passed around the logical ring is called the token rotation time. Token-passing networks can be extended to two kilometers (some as high as eight kilometers) with fiber-optic cable.

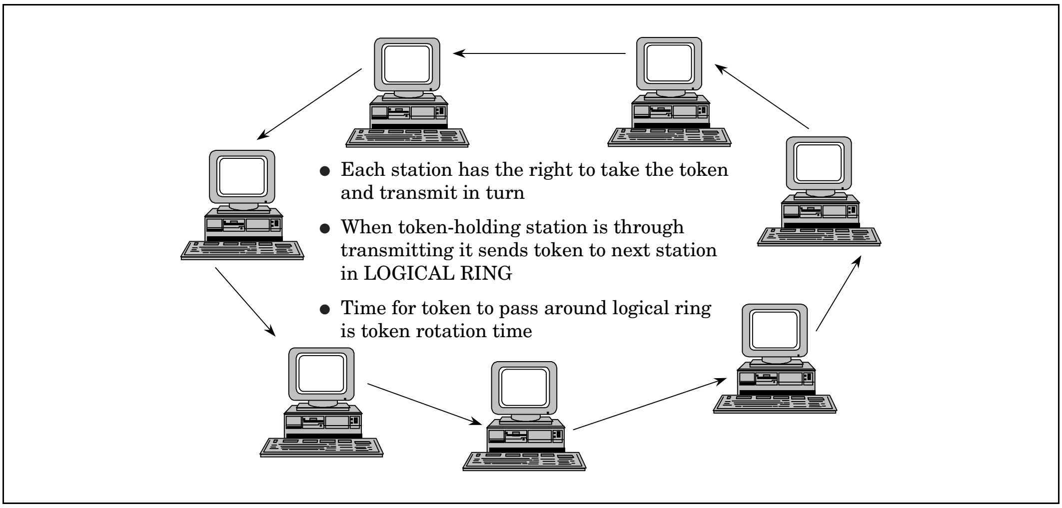


Figure 24-6. Endorsed by IBM, Proliferating Worldwide, and Defined by IEEE 802.5 Standard

Token passing—In digital communication, a media access method (ISO data Link layer 2) for peer-to-peer communication bus nodes. The token is passed around to each node on a regular schedule, making throughput and response time predictable (deterministic). Only one node has the token at any one time, and during that time it has complete control of the network.

The Conquest of Speed

A common mistake when measuring the speed of transmission is to be caught up the "specsmanship" game of proclaiming how many bits per second are operating on that network. There are many more factors than merely bits per second. The real test is how much *information flow* can be operated over that network. As an example, we can compare electronics to our analogous telegrapher from the beginning of Chapter 22, who sent every letter of every word of every sentence of every paragraph to communicate an idea or sent two-letter codes that can convey the same idea without having to use all the letters. The example we used was "LT" (look at the tank) or "CW" (check the water). The same thing applies to electronics. There are key "shorthand notes" within the communications structure that can move a lot more information, but that's only part of the story. Another part of it includes how to locate the specific data needed, how to convert that data from electronic processing data into digital signals, and then how to convert it back again in other devices or stations to the form it requires. With the pressure for interchangeability between vendors, this becomes quite a challenge to any attempts at innovation and product differentiation.

- Several factors must be considered in considering transaction time on the network:
  - Request message length
  - Media access time; transit time
  - Turnaround time (find, process reply)
  - Response message length
  - Transit time for return
  - Processing time of originator

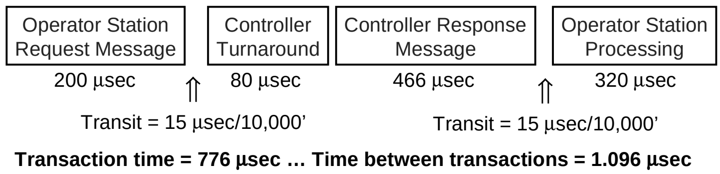


Figure 24-7. Factors in a Complete Transaction

Transaction—In communications, a transaction is a message destined for an application program, usually a computer-processed task that accomplishes a particular action or result and then responds with the result to the initiator.

Turnaround Time

One of the devices needed for communication is a modem, which transforms information from the electronics of the station into the network’s needs. Strictly speaking, a modem transforms binary data into an analog form and vise versa. Modems (an acronym for “MOdulate/DEModulate”) are usually associated with telephone line transmissions, yet the signal on line has constantly varying values. In distributed control systems (DCSs), bits are transmitted and have either an “on” or “off” state, our good old ones and zeros. However, the interfacing action between the binary information of the controller has some similarity to that performed by a modem, so the term is accepted.

Amplitude. Height of alternating or oscillating signal.

Modulation. Process by which the characteristic of one wave (the carrier) is modified by another wave (the signal), such AM, FM, and PCM. See *AM* and *PCM*.

AM—(Amplitude Modulation). Transmission technique in which the amplitude of the carrier is varied in accordance with the signal. See *PM* and *FSK*.

FM—(Frequency Modulation). Method of transmission in which carrier frequency varies in accordance with signal

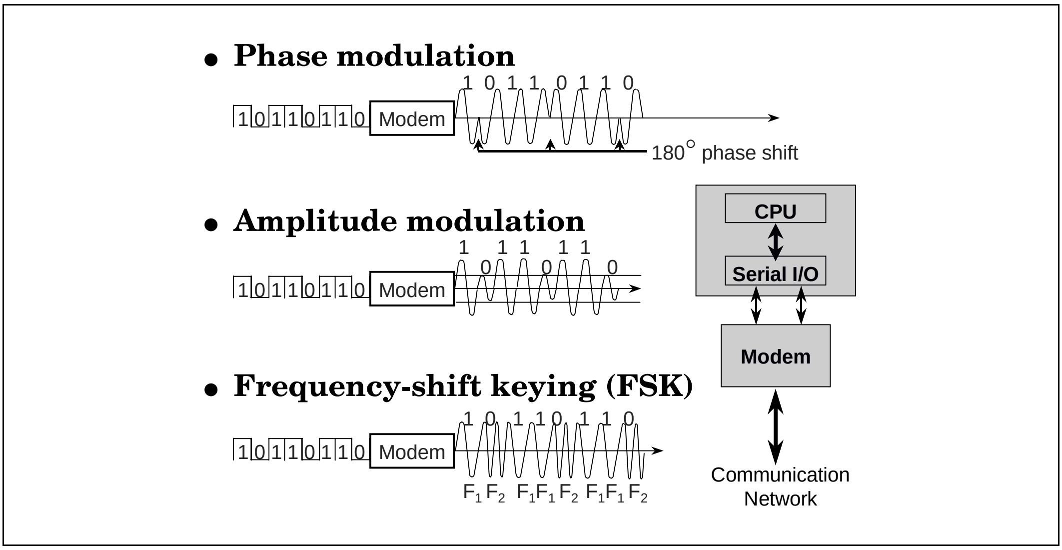


Figure 24-8. Three of the Ways Ones and Zeros Are Transmitted.

FSK—(Frequency Shift Keying). Method of data transmission using frequencies to indicate the state of the bit being transmitted. See *AM* and *PM*.

PCM—(Pulse Coded Modulation). Technique in which an analog signal, such as a process sensor signal or voice, is converted into a digital signal by sampling the signal's amplitude (slicing) and expressing the different amplitudes as binary numbers; the sampling rate must be twice the highest frequency in the signal. See *PM* and *FSK*.

PM—(Phase Modulation). Method of transmission whereby the angle of phase of the carrier wave is varied in accordance with the signal. See *PM* and *FSK*. One of three ways of modifying a sine wave signal to make it "carry" information; the sine wave or "carrier" has its phase changed in accordance with information to be transmitted.

Modems use three basic modulation schemes in general. The names of these schemes are frequency-shift keying (FSK), amplitude modulation (AM), and phase modulation (PM). FSK is a form of frequency modulation. A carrier frequency is changed to one value representing "1" and another representing "0." AM operates the carrier at a constant frequency that changes the value of the amplitude to correspond to the changes in the state. PM shifts a transmitted signal's phase by a specific number of degrees corresponding to an incoming bit pattern. Obviously, all three methods can handle a number of levels of information, not just "on" and "off" (1s and 0s). Distributed control needs only this "simple" data communication, and because FSK has some signal-to-noise advantage, it has become the most commonly acceptable form used for these purposes.

In addition, the carrier can be unchanging when no data is being transmitted at some neutral voltage. When a message is about to be sent, a signal varying from a positive to a negative voltage with respect to a neutral level might be sent at a frequency of, say, 2 MHz. After a predetermined number of variations, the message would follow. To send the message, a carrier would vary its voltage from positive to negative at the rate of maybe a half megahertz for a logical one and a one megahertz for a logical zero. After the transmission is complete, another burst of signal at a different frequency might indicate the end of the message, after which the line would go back to its neutral condition.

To get information out of storage registers and onto the port interfacing on the network, there will be some sort of Serial Input Output (SIO) chip that would change the parallel format of the bits into a serial one. By the mid 1990s the chip probably included a universal asynchronous receiver-transmitter (UART). This device is made up of shift registers into which information comes from a station in parallel form and out of which it is strobed by a clock signal, one bit at a time, to put it into a serial arrangement. Along with this, there is circuitry that packs into the message "packet" bits for start, end, parity, and checking information. At the receiving end the bits that are not part of the message (the "overhead") are stripped out, and the message bits are put back into registers in parallel form. Modems can function as both transmitters and receivers, and they must readjust their circuitry after each transmission to perform the opposite function. All of this activity contributes to the "turnaround time," which is a limitation on the useful information that can be transmitted at any given period. Speed then, involves not just how many bits per second or how efficiently you can frame these bits to communicate more information, but how quickly you can convert these bits into useful information at either end of the message, or the turnaround time. There is, of course, more to our story.

Media Access Time

Probably the least length of time in a transaction is the time it takes for the messages to physically move down the communication media. Picture, if you will, not balls in a pipe, as some physics books render it, but rather flashes of light, as you would see if you were communicating a signal like the Morse code with a flashlight. How quickly can the other side respond to these flashes, and how many of these flashes can you put into the system? One must consider the load of the system, which is how much information or bits are being pushed in, and its ability to respond to that load, which is how well you can recognize the distinction between the energy levels.

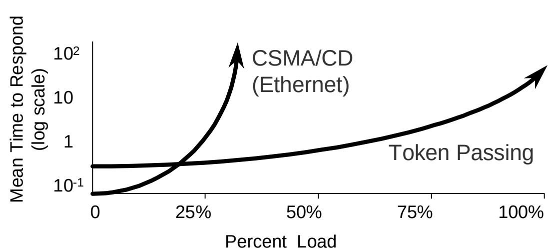


Figure 24-9. Ethernet versus Token Bus

Let us compare two common access protocols. CSMA/CD, or Ethernet, has a very fast mean time to response for the first 25 percent of its load, but then, as the load gets heavier, the curve goes asymptotically high, when it climbs to almost 50 percent of the load, so it cannot respond to real-time needs after it is only half loaded. The trick on an Ethernet system in process control conditions (to maintain "real-enough-time") is to reduce the number of stations or the distance traveled, so that all the messages can be tolerated without being restrictive. A token-passing scheme, however, requires a little more mean time to respond at lower loads, but it is more capable of maintaining a reasonable response time through the entire 100 percent of load. Clearly, it does not have the same time characteristics.

Ethernet is typically nonredundant, nondeterministic, and load dependent. This is overcome by keeping the load under 25 percent by increasing speed, limiting the number of stations, and creative messaging. It runs into the danger of saturation when loading goes over 25 percent. This can happen when traffic suddenly increases, such as during a sudden burst of alarms, which at best will slow down the response time and at worst will cause a crash. One vendor assures its customers that they never exceed 17 percent of load. This means that they only work to 17 percent capacity of the full stated bandwidth of the network [usually stated in Megabits per second (MBPS)]. This is rapidly changing, however; see "Overcoming the Shortcomings," below.

Token bus is deterministic and can manage full load but also needs similar restrictions to stay in the real-time operating realm. Nevertheless, because it cycles at a fixed rate it has a better capability to withstand an alarm burst or similar traffic increase with no loss of performance, or lockup. This system is likely to operate over the full bandwidth specified for the system.

In another example, a communication system could have a higher communication rate in bits per second but much less information flow. A token *ring* requires a sending station to pass its request for information to each of the stations in the ring before the reply returns back to itself. Whereas in a token *bus* arrangement when each station has the token it can talk to any of the other stations and get its reply in a direct response to the transaction without waiting for the token to pass around the entire ring. Both are token passing, but it makes a significant difference in how the communication moves. A carrier insertion protocol access is similar to the token ring, in that while functionally different, and also operating on a bus, it still requires waiting until the signal comes back after visiting each station.

Future of Ethernet

Ethernet is rapidly becoming the preferred control network technology in new systems emerging during the late 1990s. Ethernet will be the area network, the control network, and the I/O network. In five years, it will be everywhere. The Ethernet that is used in these applications, however, will likely be slightly different than the Ethernet of today. Advances like the 1 Gigabyte Ethernet, redundancy, quality of service, and the like will make the future Ethernet much more effective, cheaper, and powerful than

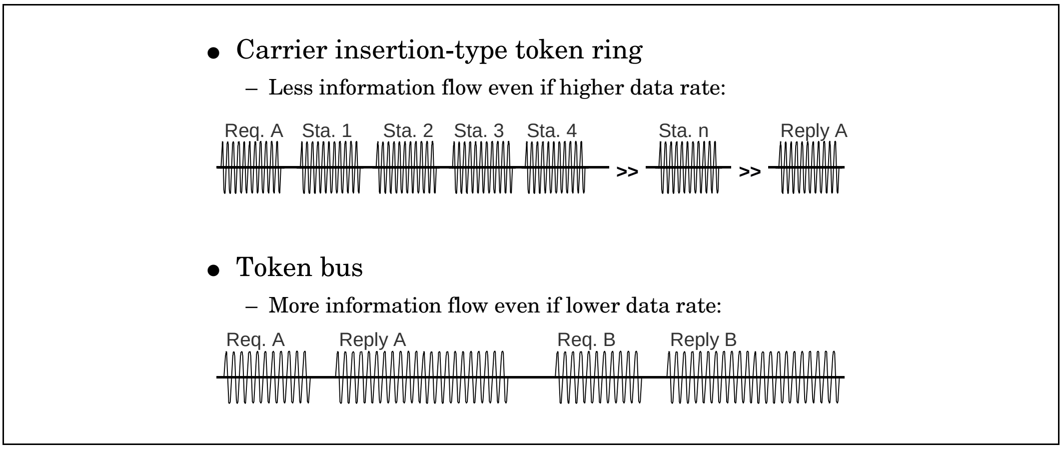


Figure 24-10. Pony Express Rider versus the Business Meeting

any proprietary control or I/O network available today. This is not fully here in1998, but it is coming. Not all the vendors have the newer technology at this point, but some vendors are already trying to cash in on it. The later releases may, however, have newer enhancements.

IEEE (Institute of Electrical and Electronics Engineers). International professional society that issues its own standards and is a member of ANSI and ISO.

IEEE 802.3—Ethernet Local Area Network (LAN) standard except for type field (10 Mbps); physical-layer standard using CSMA/CD access method on bus-topology LAN.

IEEE 802.4—Physical-layer standard using token bus-topology LAN; nearly identical to Manufacturing Automation Protocol (MAP).

IEEE 802.5—Physical-layer standard using token ring-topology LAN.

IEEE 802.6—Metropolitan Area Network (MAN) or High Speed Local Network (HSLN) standard.

IEEE 802.11—Radio and wireless LAN.

IEEE 802.12—Draft standard for 100BASE VG networking.

Ethernet Advantages

This ubiquitous network technology is sold in high volume all over the world. Towards the end of the 1990s, PC boards for this are $50 compared to $900 or more for control or device-network PC boards. For cost-conscious users and the suppliers who must meet their needs there is no contest. The fast growing development of information technology (IT) is causing the abundant use of Windows NT devices and other PCs with Ethernet ports and drivers as standard (no extra cost). The easy connectivity of the Ethernet to the Internet is another factor. The Ethernet is another technology rapidly finding its way into the reality of factory and process automation, working inexorably toward sensor-to-boardroom integration. Expect to see the Ethernet also work its way into fieldbus networks because of its speed and bandwidth in addition to its significant availability and cost advantage.

Overcoming the Shortcomings

Several IEEE Ethernet standards and standard enhancements released during 1997 and 1998 to address three main areas of performance “pucker-points”: transmission speed, determinism, and reliability.

- Reducing collisions is the goal of IEEE 802.1 p for traffic class expediting or message prioritization. Initially designed for multimedia applications, The standard enables system designers to prioritize messages, guaranteeing the delivery of time-critical data with deterministic response time and repeatable results. The standard should rapidly replace those control vendors who have made proprietary modifications on their own to achieve this effect.
- Reliability is the goal of IEEE 802.12d for redundant links. Adding redundant links to a network allows the automatic recovery of network connectivity when a link or repeater failure occurs anywhere in the network pathway. As a standard, this should also eliminate the several custom solutions which have been created by different control system vendors.
- The emergence of 100 Mbps bandwidth has been enhanced by two additional specifications to boost performance through increased bandwidth. IEEE 802.3x, for full duplex, allows bi-directional, simultaneous transmission and reception of standard Ethernet frames using separate transmit and receive channels. IEEE 802.3z, for Gigabit Ethernet, will allow a factor of ten-times-faster-transfer of Ethernet format frames while maintaining maximum compatibility with the installed base.

Let’s Not Get Carried Away

No one is suggesting that we really connect the business systems *directly* to the plant floor! Not only could this be dangerous, it overlooks the fundamental purposes of a navigational control systems requirements compared to those of the transactional business system. There are significant capacity and speed differences, loading into the battle of “What is real-time?” as opposed to “real-enough time.”

Using Ethernet in both realms will be best served by *separating* these domains with the following standard network technologies:

- **Bridges—**which are the simplest technology for this function, connecting network segments which can be different physical layer types, such as Ethernet and FDDI; while they receive all signals, by reading addresses they can selectively determine the appropriate segment to which it will pass a signal (such as not sending it to the originating segment again), thereby increasing network to maximum possible size.
- **Routers—**use logical and physical addressing to connect two or more logically separated networks by organizing a large network into subnetworks with their own

logical address; functioning similar to bridges, they also keep the networks separated, but with access to each other, hence reducing traffic in each.

- **Hubs**—or dumb hubs, can use passive hubs to extend network to longer distances, using the connection of several star topologies, sending all messages to all stations, thereby improving where stations can be located; active hubs act as repeaters, sometimes called multiport repeaters, either amplifying the signals (noise & all) or regenerate signals (replicate signals without the noise). [Figures 24-11 vs. Figure 24-12]

- **Smart Hubs**—also called intelligent hubs, managed hubs, or switched hubs, allow connection of more than two networks by looking at the destination address and forwarding the message to the proper network segment; thus, by limiting the number of signals along each leg, reduces traffic on each, which reduces the potential of collisions, and improves performance [Figures 24-13 and 24-14]. One user's experience changed a 50% collision rate to only 1% by using smart hubs in a large system.

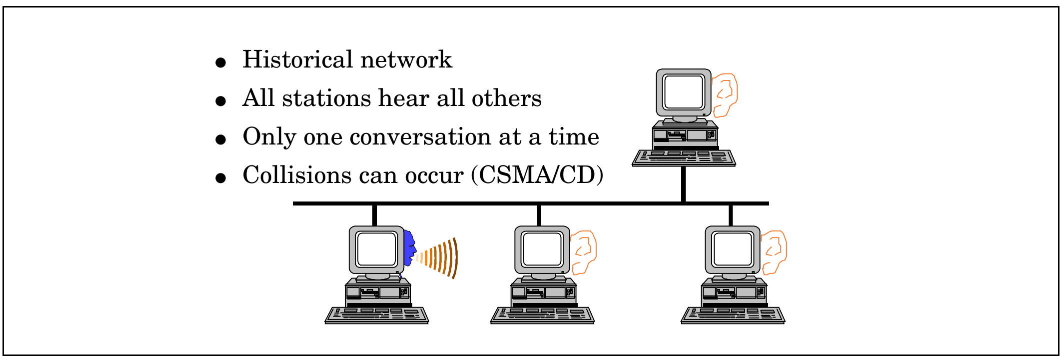


Figure 24-11. Traditional Ethernet

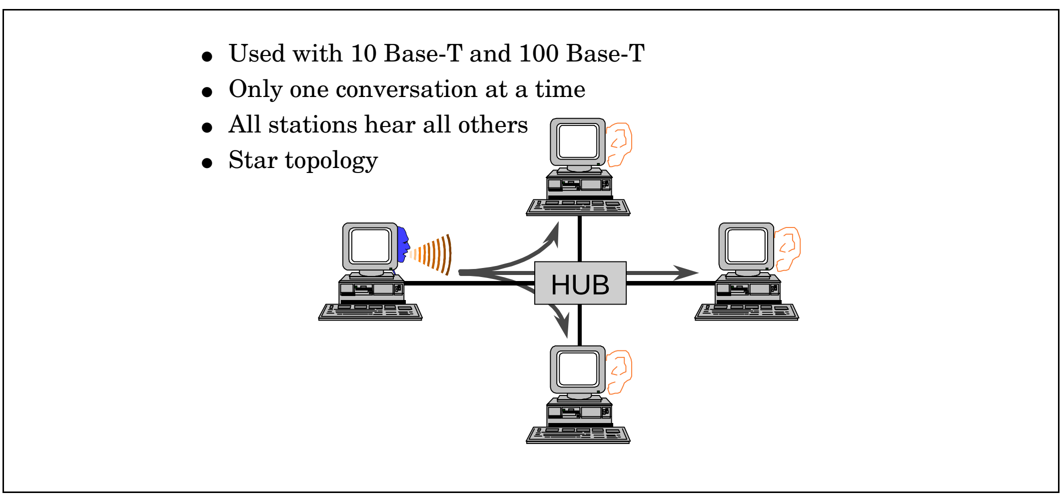


Figure 24-12. Hubs Can Allow Individual Station to be Isolated from Network

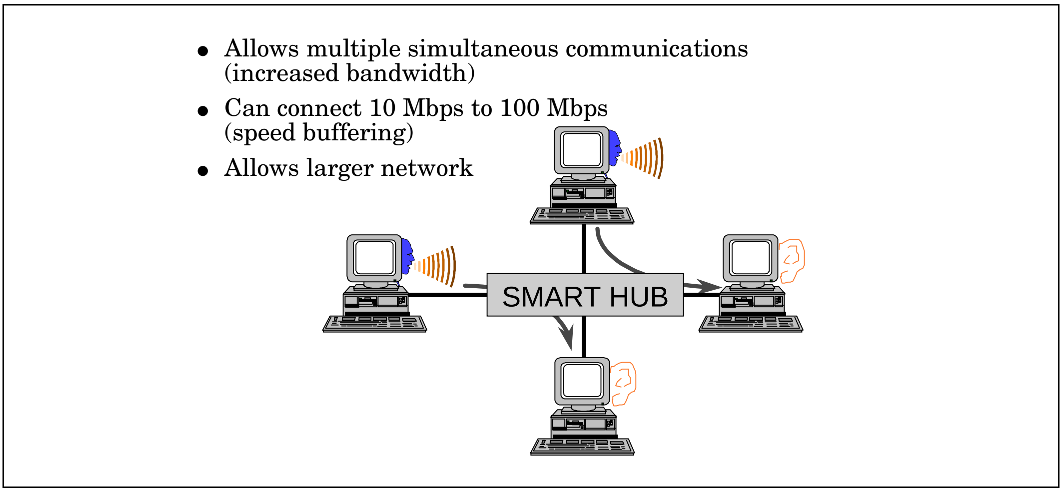


Figure 24-13. Smart Hub "Knows" Network Layout so it can Route Messages to Proper Port

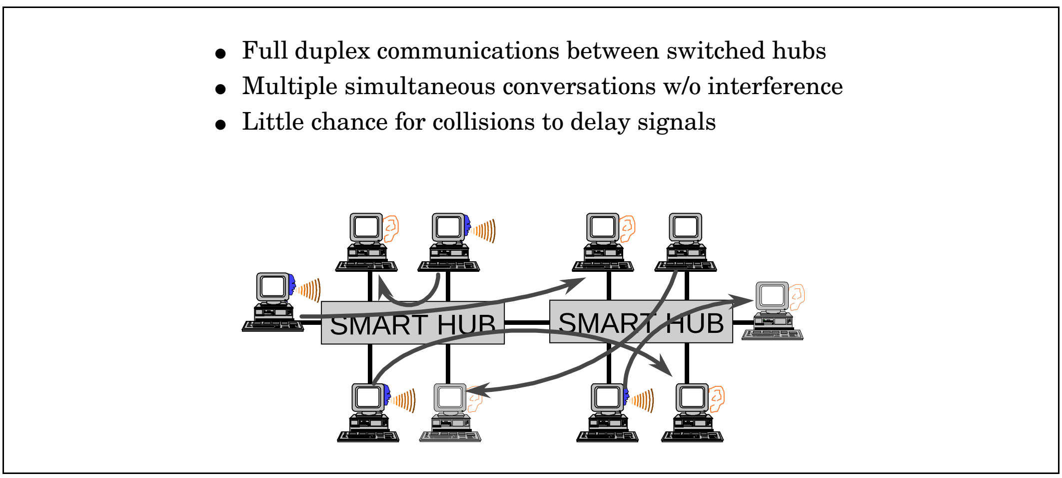


Figure 24-14. Smart Hubs in Networks Allow Any Two Stations to Talk Without Interference

Emerging Technologies of the Late 1990s

As the 1990s come to a close, there are already some other technologies with the potential for application in factory automation and process control. These are showing up in some business systems but are not yet being used by any control system suppliers at the control level:

- **Asynchronous Transfer Mode (ATM).** A physical layer protocol that performs the functions of a network layer protocol, ATM was designed for use in broadband ISDN (Integrated Services Digital Network) systems to carry voice, video, and data simultaneously. Lacking the support structure of Ethernet and somewhat expensive, it could compete with Gigabit Ethernet if volume increases.

- **Universal Serial Bus (USB).** A serial bus protocol developed by a consortium of computer companies as a single replacement for the collection of serial and parallel ports on PCs for the many peripherals (keyboard, mouse, printer, scanner, etc.). Already being shipped in PCs, cost-for-volume, speed (12 Mbps), and deterministic features give it potential.

- **IEEE 1394 (Firewire).** A serial bus protocol with high speed (>100 Mbps) for multimedia peripherals external to the PC chassis, such as digital video camcorders, video conferencing cameras, high-speed disk drives, and the like. Firewire holds promise its the speed and bandwidth, assuming volume brings down the price.

Compatible versus Compliant Standards

Many vendors have confused users in their rush to sound like all things to all people. In discussing their architectures, they frequently declare they are “compatible” with a specific standard. The user hears or thinks that means they are “compliant” to that standard and that the system will work interchangeably with any other system that is “compatible/compliant” with/to the same standard. There is a significant difference.

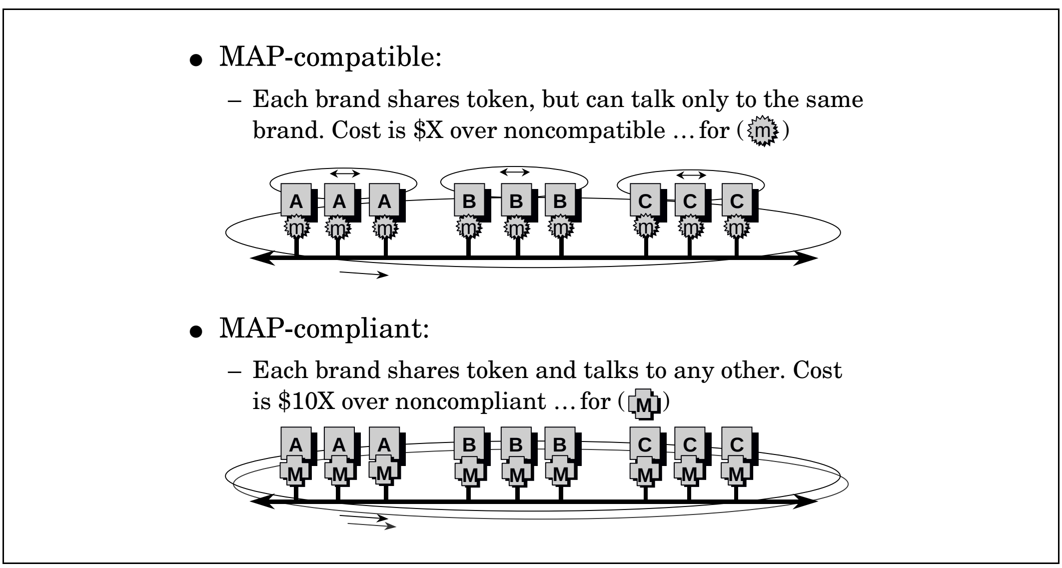


Figure 24-15. Many Have Been Confused by Misuse of This Terminology

Compliant—Conforms exactly with rules of a recognized standard. For example, a specification-*compliant* device will operate on a standard communication system and communicate with all other devices on that system made by any other vendor who is similarly compliant.

Compatible—Can coexist with the rules of a recognized standard, but may not be compliant. For example, a specification-*compatible* device will operate on a standard communication system and communicate with other devices on that system made by the same vendor and will not interfere with devices made by other vendors on that same system, but it will not be able to communicate with those other devices.

MAP (Manufacturing Automation Protocol). Based on IEEE 802.4; General Motors originated this networking protocol which follows seven--layer OSI model (Note: MAP-compliant means it conforms to specifications and will talk with other devices over the system. MAP-compatible means only that it will not interfere (physical and data link layers only) with "foreign" devices, but will only talk with like devices while sharing the timing with those foreign devices.)

MAP/EPA (MAP Extended Performance Architecture). Dual architecture that supports both full MAP seven-layer communication architecture as well as the architecture for time-critical communication that bypasses Layers 3, 4, 5, and 6.

Open Communications Standards 25

"I think standardization encourages creativity! It provides the framework upon which new developments can be conceived and built. For example, without a dictionary and rules of grammar to provide the standards for words and their use, a complex language cannot evolve, without which a complex written literature cannot be cultivated. In the great civilizations, a higher and higher standard of living requires more and more standardization, which produces both brilliant and beautiful improvements...

...which we can all later take for granted, of course!"

Marilyn vos Savant—April 27,1997

Open Systems: What Are They?

Open systems are based upon *widely accepted* industry standards. The question is who makes the standard become standard? To answer this we must consider the two types of standards:

- *Regulatory* or *consensus* standards, which emerge slowly and painfully from technical societies such as ISA (the society), ISO, IEEE, EPRI, ANSI, and hundreds of others. Fortunately, many of these societies are teaming up to avoid confusing and ambiguous overlaps. A strong international effort is at work by many enthusiastic individuals to homogenize many existing and newly required standards. Delays in the emergence of standards in the past came from necessary engineering thoroughness probably more than (from) individual egos. The biggest delaying factor by far, however, is the time for communication of ideas, the time for reviews, the need for many engineers to travel great distances to meet face to face, and the need to convert many good spoken ideas onto paper by people with little spare time. Although supported by industry, this effort itself is done almost exclusively by volunteers! Some exciting efforts using e-mail, the Internet, conference calls, and even "computer conferencing" have significantly reduced communication and publishing delays. Past examples of this include the OSI model from the ISO, POSIX, and FF Fieldbus.

- Often by virtue of *market dominance de facto* standards avoid these messy problems by the fiat of the largest manufacturer, by their sheer popularity with users, or sometimes by the accident a of combination of circumstances. The drawbacks involved in waiting for this seemingly more efficient standards = a creation method include having more inferior products because a company has better marketing than engineering or not having any standards at all because the market is saturated with several equally popular products. Serious delays in regulatory standards will cause impatience in the industry and encourage many vendors to race each other for the de facto prize. Sometimes a user company that is large enough to influence an industry will declare a standard. When General Motors announced the need for MAP, and refused to purchase anything not fitting that standard, they began to discover that many functions did not fit that standard either! Even though good things came from the effort, the resulting fragmentation of the standard so as to overcome the deficiencies weakened the end result considerably. In fact, it turned many people off of the idea of the possibility of sweeping worldwide standards. Past examples of this type of standard include ISA (the IBM® standard), FORTRAN, MS-DOS, UNIX and SQL.

(Note: actually, MAP is a *specification*, not a standard!)

ANSI—(American National Standards Institute); nonprofit, independent organization supported by trade organizations, industry, and professional societies for standards development and coordination in USA; they represent USA to ISO; they defined ASCII.

EPRI—(Electric Power Research Institute); research consortium of 660 member utilities in U.S.

FF—Fieldbus Foundation, a not-for-profit organization dedicated to developing single, worldwide, interoperable fieldbus... formed from merger of WorldFIP North America and ISP Foundation in 1994.

FORTRAN—FORmula TRANslator; first high level computer language, developed by IBM® (1954); known as scientific language because of its facility for “number crunching,” solving engineering, mathematical, and other scientific problems; procedure-oriented and has good array handling features.

IEEE—(Institute of Electrical and Electronic Engineers); international professional society that issues its own standards and is member of ANSI and ISO.

ISA—(Instrument Society of America); organization now formally known as “ISA, the international society for measurement and control,” started in1945 in U.S. which is made up of individual volunteers from all aspects of the instrumentation business, representing vendors and users, many types of industries, and nearly every job description; they traditionally provide education and develop international *consensus* standards and practices; BUT also means: Industry Standard Architecture, a *de facto* personal computer 24-bit bus standard used in IBM®PCs and compatibles; developed for extension cards in the first IBM PC, it originally supported only 8-bit wide data path (now called PC/XT bus); subsequently developed to 16-bit for the AT class computers, and called AT bus, supporting both 8- and 16-bit cards.

ISO—(International Standards Organization); international organization for promoting the development of standards for computers.

OSI—Open Systems Interconnect; 7-layer reference model for network operations standardized within ISO to enable any two OSI-Compliant devices to exchange information; the seven layers are Physical, Datalink, Network, Transport, Session, Presentation, Application.

POSIX—Portable Operating System Interface eXchange; originated for computer environments as means of standardizing critical interfaces for the many divergent variations of UNIX® operating system; living under auspices of both IEEE and ISO, it has evolved into an entire family of standard interface definitions, no longer limited to UNIX®; specifies how software applications and operating system software should be implemented so that applications can be ported to other POSIX compliant environments.

MS-DOS®—Microsoft Disk Operating System; developed for IBM® PC and has become a de facto standard; sometimes called PC-DOS designed to control and manage I/O devices and memory for personal computers (PCs).

SQL—Structured Query Language [pronounced "see quill"]; ISO database access standard for communicating (querying, updating, and managing) with various relational data bases; allows client to access only that data required to satisfy a specific request, reducing network traffic and improving performance; derived from an IBM® research project that created Structured English Query Language in the 1970s; now accepted standard in database products.

UNIX®—UNIfied multitasking; time-sharing operating system designed originally by (and trademark of) AT&T for communicating, multi-user, 32-bit minicomputers which has become widely accepted because of its versatility; tradename now belongs to X/Open Corp. Ltd.

Examples of standards in everyday life include automobiles, plumbing supplies, electrical service, audio and videotapes and the like. Many spare parts are interchangeable, but frequently there are several versions in any one industry. Sometimes this is due to the existence of a variety of equally influential manufacturers, sometimes to the need for functional differences, and sometimes because of inaccessibility, such as international distribution.

open system—Hardware/software designs in which a degree of interchangeability and connectivity provides the user with choices: the ability to select multiple products from multiple vendors and integrate them seamlessly on powerful networks. Open systems make every resource on the network available to any authorized user who needs it.

Why Are Open Systems Important?

- Why should a company care about openness in a control system? Open systems are important to you for the following reasons:
- You are in a competing global marketplace, and you cannot fall behind your competition. Generally, with the many emerging capabilities of such a variety of different hardware and software products, your company must be nimble enough to add more functions to quickly meet your customers, rapidly changing requirements.
- You must continue to improve your plant operations year after year; otherwise, you will risk falling behind your competition. New methods of production are arising very quickly. How easy is it to add improvements to your way of producing?

- No one vendor can meet all your automation needs, you must rely on multiple vendors. Although the industry has not yet reached the point of “swapping out” a system controller with one from another vendor, there already exist circumstances where one vendor’s workstation is capable of residing on a network and operating controllers from several other vendors. (That station, however, cannot *configure* those controllers.)
- To manage your plant operations effectively in response to fluctuating demand for your products, you will have to network all the automation systems together. This will require a degree of interconnectivity through communications standards.
- Computer technology will continue to advance at a breathtaking pace and you cannot afford to avoid using it. It will become even more important to make allowance for regular upgrading, but with no idea, at the time you purchase, *which* parts will need expansion. Your competition will always be adding a growing list of tools.

- You want to preserve your existing investment in computer software as you upgrade your computers. Software will comprise nearly three-quarters of the total cost of automation systems. It better work together.

The Most Common Definition: “My standard is what I have installed”

This definition only works if

- Your vendor can *alone* provide you with cost-effective solutions to your current and future needs.
- Your vendor has the financial resources to be innovative and keep pace with technology in *every area* of process control.
- Your vendor is committed to being compliant with industry standards as they emerge.

It costs vendors hundreds of thousands of dollars to participate in an emerging standard! No vendor “can do it all.” Interoperability and the interchangeability of systems are essential for a system user to survive. This may not be entirely possible just yet, and the concept will still be a “journey toward” for some time to come. While there is nothing wrong with the economies provided by “single sourcing,” that should never be confused with allowing yourself to be “held hostage” by the vendor.

Advantages and Disadvantages

Open systems preserve user/vendor investment through the following (Figure 25-1):

- **Applications portability.** The ability to use the same application software on computers from different vendors.
- **Vendor independence.** The ability to have computers from different vendors work together.

- **Scalability.** The capability to use the same software environment on all classes of computers, from mainframes to PCs.
- **Personnel portability.** Personnel can be easily trained and moved from one system to another.
- **Evolution of solutions.** User investment is preserved through the planned progression of standards.

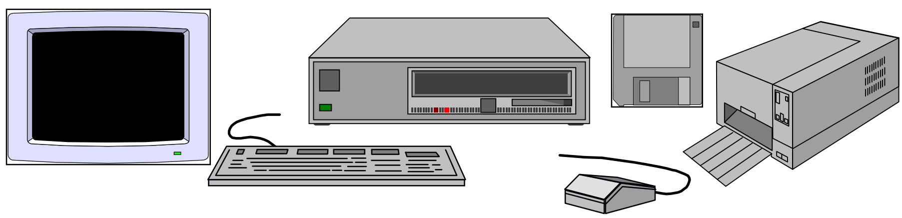


Figure 25-1. Without Standards, Users Would Have No Choices for Equipment, Components, or Supplies for Them

The drawbacks of industry standards include the following:

- Consensus standards can take years before they are fully defined and implemented by the vendors. Pressures from users are real here and have made differences.
- Standards are in general very comprehensive in nature. As a result, products tend to be more expensive, especially in the early stages. Once accepted, however, they can become very cost effective, sometimes not in their actual price but because of the versatility they offer.
- Standards may hold back some technological innovation. The pressure will be on vendors to be creative *within* the requirements of a standard. This has worked in other areas in the past.

Few believed that 4 to 20 mA and 1 to 5 Vdc signals would ever become a standard. By the way, *that* took about a quarter century and was essentially forced by users.

Some of the advantages just described may look quite idealistic. De facto Microsoft standards have opened capabilities in the office world so much that “big-time computing” is now available in the smallest of offices. This same technology is having a similar impact in changing automated control system in ways unheard of even in the recent past. It is moving so fast that readers of this book will probably already have seen examples that the author could only guess about while writing these words.

Fallacy About Standards

A major fallacy surrounding standards is that: "Products based on industry standards are identical." This is a confusion involving the difference between specification and implementation:

- A standard is a commonly accepted *specification* of a set of rules
- *Implementation* is how a specification is made into a reality

The reality behind standards is quite different:

- Products based on an industry standard are not identical.
- Major vendors will continue to implement standards uniquely so as to provide product differentiation.

As endless variety of features abound on home televisions. Yet they must adhere to reception standards, safety standards, quality standards, and so on. Just look at how many television *tuner* variations there are, all of which work on any brand! In the automotive world, all vehicles must adhere to a wide variety if national and local standards for operability, clearances, and driving procedures (steering wheel position, turn signals, mirror positions, tire sizes, and so on). However, most every driver in the world, across incredible culture boundaries, knows "PRND321"! (Look at your automatic gear shift lever).

The use of standards increases competition from smaller vendors, who can supply "niche" products. In the distributed control systems (DCS) market, several systems can consist of parts from different vendors.

Often if one vendor makes a product obsolete, another vendor's product can replace it easily. That second vendor will have reduced design costs, because

1. It does not need to provide the entire solution, only what it does best.
2. It can build on standards; it does not need to reinvent *everything* (e.g., a software vendor does not need to rewrite DOS).
3. Meanwhile, the user can select "best-of-class" equipment from different vendors and integrate the components into a system that is, perhaps, superior to what is available from any one vendor.

Developing Standardized Links: The Ubiquitous RS-232

With regard to interconnecting, that is heterogeneous systems that had computers or microprocessors in them, the RS-232 Serial Link has been the predominant means employed. Although it was a "standard" (from the Electronics Industry Association) it fell short. Even if two vendors touted RS-232 conductivity, they often could not communicate with each other without requiring the development of special software drivers or custom cables. One vendor might support four wires, while the other used seven. (The standard specifies twenty-five signals, many of which are optional.) Each vendor could define its own communications protocol. The problem was that the standard dealt with only one layer of the seven-layer Open System Inter connection (OSI) model. And that was the physical link layer. So, of course, this gets far more complicated when you deal with the other six layers, which compounds the problem far beyond what people anticipated with the so-called openness of the RS-232.

Communications Structure: The Open Systems Interconnect Model

Just from our brief discussion so far, you can see that there have been many kinds of protocols used for different circumstances in communication. An effort has been made to organize the hardware and software tasks of networking, through the International Organization for Standardization (ISO), which has developed a reference model for the protocols used in communications. Its formal name is the "International Organization for Standardization Reference Model for Open Systems Interconnection (ISO/OSI)." In their use here, open refers to communication systems that have provisions for interfacing with other nonproprietary systems, using established interface standards (Figure 25-2). The ISO model categorizes the various protocols into seven layers, each of which can be involved in transmitting a message from one system element to another (Figure 25-3). We will attempt to describe these layers will be done with an analogy to the postal system and its equivalent function. Hopefully, this will make it more understandable to the uninitiated. We will describe these layers in reverse order:

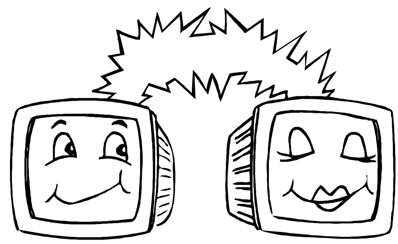


Figure 25-2. The OSI Model Allows Communications Among Computers from Many Different Suppliers Communicate Over Several Different Networks!

- **Layer 7: Application.** This layer really isn't part of the communications protocol structure but rather is part of the application software or firmware that is transmitted *through* these lower layers. In a higher-level language program, for example, it could be a statement that requests information from another system element over this communications media. In a function block structure it could be an input block, for example, that requests that certain process variables be read from another station over the communications system. In our postal system analogy (in Table 25-1), the application is really the contents of a letter inside of an envelope.

- **Layer 6: Presentation.** This restructures the data through some standardized format used within the network. It translates the message formats in the communication system to the information format required by the application layer (layer 7). This allows the application layer to properly interpret the data sent over the system, and, conversely, it puts the information to be transmitted into the proper message format. The postal system equivalent of it is the format and language style of our letter, including the translation of the language that's required, say, from American to German.

- **Layer 5: Session.** This name and address translation acts as security in the synchronized and managed data, which schedules the starting and stopping of communication activity between two elements in a system. It also specifies the quality of transport service required if multiple levels of service are available. The postal system equivalent is the name, address, and zip code of both the receiver and the sender, as well as the postage stamp determining the class (air, land, bulk rate).

- **Layer 4: Transport.** This provides transparent, reliable data transfer from end node to end node. It's the mechanism in each communicating element that ensures that the end-to-end message transmission has been accomplished properly. Services provided by the transport layer include acknowledging messages, detecting end-to-end message errors, retransmitting the messages, prioritizing the messages, and transferring messages to multiple receivers. The postal system equivalent is certified or registered mail, which provide verification to the sender that the letter arrived at the correct destination in good condition.

- **Layer 3: Network.** This performs the message routing for data transfer between stations not in the same network. Within a network having multiple pathways between the elements, this layer handles the routing of messages from one element to another. In a communications system consisting of multiple subnetworks, this layer handles the translation of addresses and routing of information from one of these sub--networks to another. In communications systems consisting of a single network and having only a single pathway between the elements, this layer is generally not required. This is the equivalent in the postal system of the distribution system of transferring a letter outside the postal system to a system in another city or country, instead of within the same postal zone.

- **Layer 2: Data link.** This provides the means to establish, maintain, and release the link between systems, to transferring the data frame between the nodes in the same network, and to detect and correct errors. This layer allows access from one station on a communications network to another, so that it can be determined which station has control of the hardware at any given time and when to transmit its messages or request information from the other. This layer is at the bit level and defines the formatting of the bits and the bytes as well as the message itself so the arrangement makes sense both to the sender and the receiver. It also defines the error-detection and error-correction techniques used and sets up the conventions for defining the start and stop of each message. In the postal system, it's the segment that transfers a letter from a sender to a destination within the same postal system, either a distribution center or a forwarding center to receive the letter in another system. It allows the mail to move within the same zone.

- **Layer 1: Physical.** This includes physically transferring messages between the adjacent stations. It defines the electrical and mechanical characteristics of the interface between the physical communication media and the driver and receiver electronics. This includes the voltage levels, the channel structure, and whether it's parallel or serial transmission, for example. It also includes the signaling and the modulation technique used by the hardware to transmit the data. In the postal system, this is the equivalent of the conveyance, whether it's the postman, the truck, the airplane or whatever.

Different vendors are now able to develop compatible products for any of the seven layers. The seven layers reside in a computer. Layer 1, Physical, is hardware for the physical connection (for example, the Ethernet port); layers 2 through 7 are software "modules."

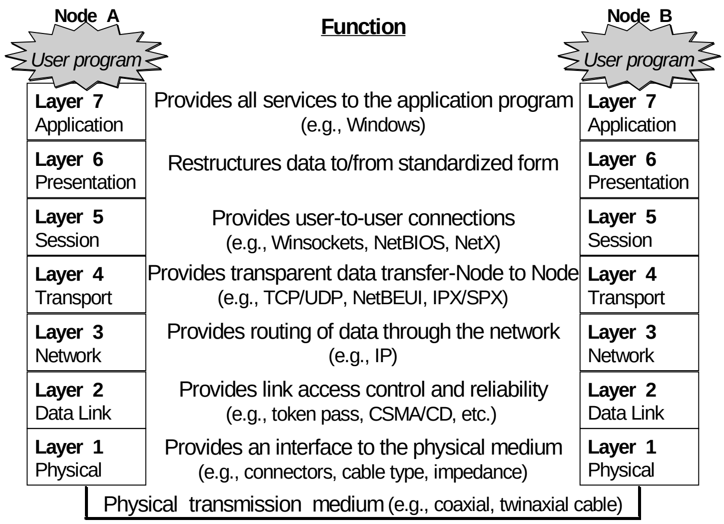


Figure 25-3. International Communications Standard

Brief Summary of the Seven-Layer OSI Reference Model

7—Provides all services directly comprehensible to application programs.

6—Restructures data to/from the standardized format used within the network.

5—Name/address translation to access security and synchronize/manage data.

4—Provides transparent, reliable data transfer from end node to end node.

3—Performs message routing for data transfer between nodes not in the same LAN.

2—Provides a means to establish, maintain and release logical data links between systems; transfers data frames between nodes in same LAN; and detects errors.

1—Encodes and physically transfers messages between adjacent nodes.

Continuing the postal analogy for all the communications layers, we want to send a letter or package to a person in another city or country. We do not need to know any of the intermediate steps; we just put the proper address information ("header") on the package and let the other layers handle the delivery details. We do not care about the tracking numbers, routing, and so on. Likewise, the receiver does not need to understand the routing to receive the letter or package. The user layer, or the program above Layer 7, is the actual information (significance of the words) in the letter content, or the function of the item being sent.

Table 25-1.
All Communications Models Come from Human Activity

OSI Layer	Postal System Equivalent
7—Application	Letter contents within envelope
6—Presentation	Format and language of letter, including proper translation into another language, if needed
5—Session	Name, address, zip code of both sender and receiver
4—Transport	Certified or registered mail; verification to sender that letter arrived at correct destination
3—Network	Distribution transfer to outside local system to another city or country
2—Data Link	Distribution within same local system, or within local system in that other city or country
1—Physical	Conveyance: postman, truck, train, plane…

As the user data moves through the seven layers, each layer adds a piece of addressing information called a header, or footer (Figure 25-4). Each layer does its job independently of the other layers, and different choices are available at each layer. Each layer's header is independent of the other layers' headers. There is no special interface required between layers.

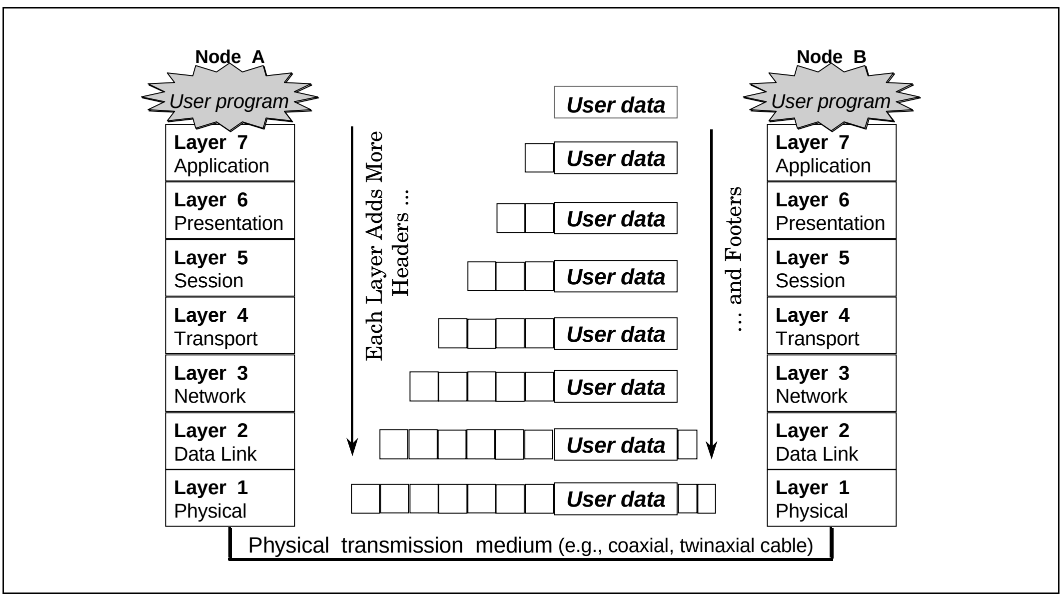


Figure 25-4. Each of the Seven Layers Adds Communications Overhead

Enhanced Performance Architecture (EPA)

EPA is a simplified OSI model for use when the network is not so complex, such as you would find in the fieldbus structure (Figure 25-5). This is divided into three areas:

- **Application services**—which is the application layer, Layer 7;
- **Transport protocol**—which encompasses the network, transport, session, and presentation layers, 3 through 6;
- **Media**—which would be Layers 1 and 2, the physical and the data link layers.

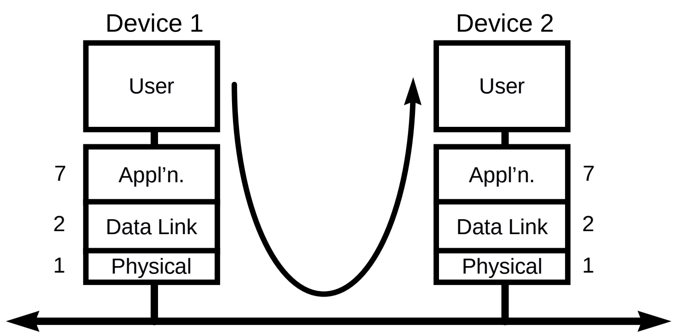


Figure 25-5. EPA Uses Subset of OSI, Cannot Cross to Other Networks, But Must Stay Within its Own

EPA enables some less complex devices on more simple communications links to be more easily defined. It is in this arena that many of the standardizing committees are working to make some accessible standards available without having to define all of seven separate sessions (layers). Most of the fieldbus efforts are here, where only digital input-output (I/O) links are defined.

In this enhanced performance architecture, or EPA, the only concern is with the user "above" the application layer, the data link, and the physical layers. Our post office analogy of the user layer is the information request that is the common language one can read and understand. The application layer is the paper that contains the message, and that is placed into the data link layer which is the envelope that addresses the to and from and carries the stamp. The physical layer is the mailbox that delivers the envelope to the media, and the physical media or the wire is the truck that transports the envelope independently of whatever message was put into it.

Shortcuts Between Networks

Each of the seven layers needs a header of code. All of these individual headers become the overhead, adding to the complexity and hence the expense, vulnerability, and delay of the transmission. Many times, however, it isn't necessary to go through all of the layers if the connected networks already have some of the functions in common (Figure 25-6). Whenever possible, “simple is better.”

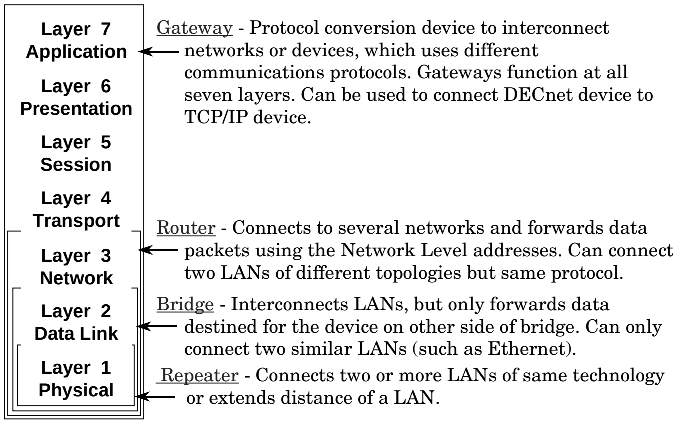


Figure 25-6. Not All Seven Layers are Needed to Connect Between Some Networks So There Are Different Connecting Devices to Accommodate Those Situations

For example, you can “bridge” across the data link layer (Layer 2) so your information only needs to convert through these two layers to reach a network of devices designed to recognize the same level of data organization and structure. A single vendor can provide its own proprietary communications network for much less cost because the bridge is built into all stations designed to operate on it.

To transfer across different networks, it is necessary to use a “router” that interconnects different networks at the transport layer (Layer 4). A “brouter” performs many of the tasks of bridges and routers without the protocol restrictions of a router, which is more expensive, complex, and difficult to install. A “gateway,” however, is needed to translate all seven layers so as to move from one system to another (generally between different vendors). Of course, that requires a much more complex (expensive) device.

gateway—A protocol conversion device used to interconnect networks or devices, which uses different communications protocols. Gateways function at all seven layers. A gateway would be used to connect a DECnet device to a TCP/IP device.

router—Connects to several networks and forwards data packets using the Network Level addresses. Can connect two LANs of different topologies but the same protocol.

bridge—Interconnects LANs but only forwards data destined for a device on the other side of the bridge. A bridge can only connect two similar LANs (Ethernet—Ethernet/ token ring—token ring).

repeater—Connects two or more LANs of the same technology or extends the distance of a LAN.

Some Other Communications Standards

- TCP/IP: Transmission Control Protocol/Internet Protocol

  By the way, it is becoming common for CSMACD (a *media access* technique) to be confused with TCP/IP (a *communication* protocol) TCP/IP has the following characteristics:

  — Developed by U.S. Department of Defense in 1974

  — A de facto standard network protocol to connect UNIX systems

  — Does not extend to physical and data link layers, hardware interface is beyond its scope. Since TCP/IP is media independent it has been implemented on variety of media.

  — It is software for Layers 3 (Network) and 4 (Transport). It handles the routing of data to the correct location and error checking to ensure that the data is received intact.

  — It can use Ethernet, token ring, or other standards at Levels 1 and 2.

  — It is the most widely used software at these levels, in large part because it comes as part of UNIX, the most widely used workstation operating system.

  — It is not necessary that the TCP/IP software come from the same vendor (but sometimes it helps).

  — **TCP:** Transport layer (Layer 4). Segments data into packets and verifies that the message got to its destination intact.

  — **IP:** Network layer (Layer 3). Routes data over network to correct LAN address.

  — **UDP** (User Datagram Protocol). Alternative to TCP that does not require the acknowledgment of messages, requires less overhead, and offers higher performance relative to TCP/IP.

- FDDI: Fiber Distributed Data Interface

  — Developed by ANSI in late 1980s

  — Similar to IEEE 802.5 token ring LAN standard

  — Physical and data link layers of OSI model (Layers 1 and 2)

  — Token pass access method at 100 Mbps

  — 200 kilometers (approximately 120 miles)

  — Often used as backbone, with bridges to Ethernet

  — CDDI (copper version) covers shorter distances, but takes advantage of installed cable systems

---

FDDI (Fiber Distributed Data Interface). ANSI standard for fiber-optic links with data rates to 100 Mbps; two 50Mbps counter-rotational token rings, synchronous, prioritized.

FDDI-II—Variant of FDDI that supports isosynchronous traffic.

CDDI (Copper Distribution Data Interface). Unshielded twisted pair, shielded twisted pair, dual-grade twisted-pair options.

isochronous—Equally timed. In data communications, timing information is transmitted on channel along with data, sending asynchronous data by synchronous means. The isochronous method involves synchronously sending asynchronous characters between each pair of start and stop bits.

asynchronous—Computer logic or communications in which all operations are triggered by a free-running signal not related to specific frequency or timing; successive stages are triggered by the completion of the preceding stage.

synchronous—Logic or communications in which all operations are controlled by clock pulses. Synchronous transmission eliminates the need for start and stop bits because everything is sent at a fixed rate.

- NetBEUI: NetBIOS Extended User Interface
  - The primary protocol used by Windows for Workgroups, supported in all of Microsoft's network products.
  - First introduced by IBM in 1985.
  - Small and efficient protocol designed for use on a departmental LAN of twenty to two-hundred workstations.
  - NetBIOS is a network Basic Input/Output System (BIOS), a high-level session-layer interface used by applications to communicate with NetBIOS compliant transports such as NetBEUI.
  - BIOS (Basic Input/Output System). Commands are used to tell the CPU how it will communicate with the rest of the computer. It contains the information typically needed upon start-up.
  - NetBIOS is a communications interface to PC-*DOS* applications.

Communicating Between Software Applications: DDE

In addition to the hardware and software needed to link networks, the actual applications must also be connected. This is especially true now that so many control and operating software packages are being developed to operate in common hardware. Often, the hardware may be common but located in different parts of the plant! Sometimes the hardware does not have all that much in common, but working between similar platforms does make this effort easier.

In Figure 25-7 we see two popular application interface options are Application Program Interface (API) and Dynamic Data Exchange (DDE). Use DDE when simple interfacing is required. DDE needs no programming, just configuration. DDE is only available for Windows but is often used for interfacing to other foreign devices, such as PLCs, when that other device has DDE Server software. No custom programming is required, only the mapping of data points. As the 1990s draw to a close, at least one major supplier of control systems for operating on NT platforms has introduced the much better performing NetDDE, which Microsoft has licensed. A still faster version has been released, and like the NetDDE it will also work with existing equipment designed for DDE.

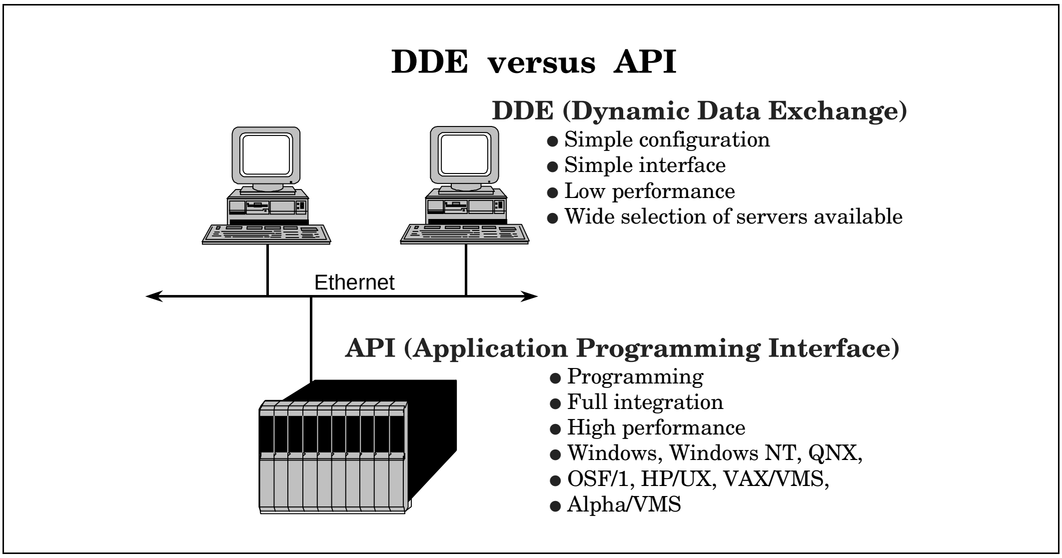


Figure 25-7. Interfacing Between Two Different Applications

An API, on the other hand, gives a more complete interface, has much higher performance, and allows full integration of databases. However, it requires programming. It is available for use with many platforms.

API (Application Program Interface). A set of formalized software calls and routines that can be referenced by some application programs to access underlying network services. Programs that use API-compliant calls can communicate with any others that use that same API. API is an interface between the applications software and the application platform.

DDE (Dynamic Data Exchange). DDE is a Microsoft developed interapplication communications protocol in which data from one program automatically updates another. Originally designed to move data from a spreadsheet to a word processor; DDE is the baseline protocol for OLE 1.0 but not for OLE 2.0 (It is supported there, however, in order to maintain upward compatibility). DDE has become more complex with the advent of Windows and WindowsNT in industrial applications, causing wags to refer to it as "Different Dynamics to Everyone."

DDE is a standard communications protocol that allows several different Windows programs to share data. DDE is a simple connection between applications; it does not understand the data itself. The *application* has the responsibility to understand the data. DDE is included with Windows.

In Figure 25-8 the DDE server is a Windows program that allows other programs to access data from the controller. It is compatible with any “DDE-aware” programs such as Lotus 1-2-3, Microsoft Excel, and Wonderware Intouch (an HMI package). The DDE client can request data, send data, and send messages over a network to the controller through the DDE server.

- **DDE:**
  - Standard communications protocol
  - Allows several programs to share data
  - Simple connection between applications
  - Does not understand the data
- **DDE server:**
  - Allows other programs to access data from controllers
  - Compatible with any “DDE-aware” programs such as Lotus 1-2-3, Microsoft Excel, and HMI software
- **DDE client:**
  - Can request data, send data, and send messages to controllers through DDE server

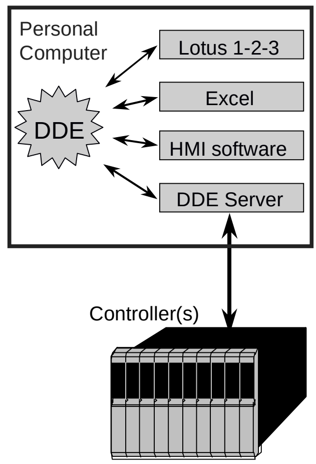


Figure 25-8. Dynamic Data Exchange Flow

Often we wish to get controller data inexpensively to a computer in the manager’s office. Purchasing an entire workstation would be too expensive if all that is needed for the manager is data for a spreadsheet but no color graphics and continuous process trend views, which are appropriate for the operator. NetDDE can help expand the data to this additional station (Figure 25-9). It was developed by Wonderware, who established an alliance with Microsoft to ensure that NetDDE is fully compatible with DDE.

In the system shown in Figure 25-8 the computer for Manager 1 gets its controller data from the controller DDE server, and the computer for Manager 2 gets its controller data from Manager 1 via NetDDE.

Communicating Between Software Applications: OPC

Object linking and embedding (OLE) was developed by Microsoft to provide component object models (COMs) that serve as the basis for integrating applications. It is, however, a very broad specification that covers hundreds of ways of sharing information among programs. There is no assurance that programs will work together just because they are listed as OLE-compliant.

A consortium of five vendors drafted a specification in 1995 to define OLE for process control (OPC). This is a standard set of interfaces, properties, and methods that applies

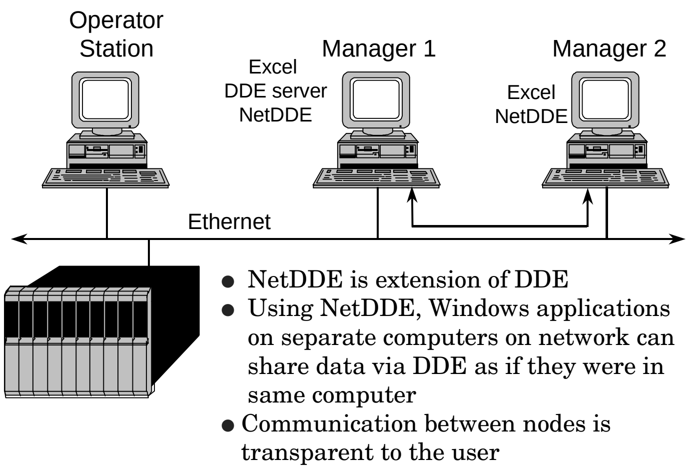


Figure 25-9. Inexpensive Sharing Data with Another Device

OLE and COM technologies to make possible greater interoperability between automation and control applications and the field systems and devices in real-time applications.

ActiveX (Active component eXtension). A Binary reusable software object that plugs into object linking and embedding (OLE) software, allowing different software packages to communicate, and interact with one another in a networked environment thus making possible plant floor integration through the Internet and intranets. Since the advent of the Internet, Microsoft has preferred to use this term over OLE because of its expanded scope (and flashier marketing).

COM (Component Object Model). In computing, a connection mechanism and protocol used to link different applications in object linking and embedding (OLE) environment. Allows the development of independent, interoperable software objects; de facto object standard by Microsoft competing with CORBA*.

CORBA—Common Object Request Broker Architecture; approach to creating open object-oriented system architectures; specifies interoperability of Object Request Brokers (ORBs); this emerging object oriented programming standard is being presented by Object Management Group (OMG); planned by 11 companies including IBM®, H-P®, Sunsoft®; competes with de facto object standard COM* by Microsoft®.

DCOM (Distributed Common Object Model). Extends common object model for the network communication of independent, interoperable software objects.

Java—Software code which can rum on multiple platforms (UNIX, OS/2, NT, MACINTOSH, etc.); originally developed by Sun Microsystems Inc. as a platform for programming on small embedded devices such as cell phones, PDAs and sensors; claimed to be more simple than C++, it has significant implications for use on Internet applications.

Java Beans—Reusable chunks of Java software code that can be assembled as components in larger applications.

OCX (OLE custom controls eXtension). Object-oriented software building blocks that save considerable programming time in the creation of applications. Theoretically, OCX can readily be plugged into Visual Basic, Visual C++, databases, spreadsheets, and word processors.

OLE (object linking and embedding). In computers, an application integration feature of Microsoft Windows and WindowsNT environments that treats data as a collection of objects to be shared by applications supporting the OLE specification. OLE enables several different applications to be linked to accomplish a given task and allows them to keep information current across several different software applications simply by changing information in one of them. With the arrival of the Internet, Microsoft now prefers to use the term ActiveX.

OLE-PC (object linking and embedding for process control). OLE extensions to improve interoperability among different industrial automation devices and software. Now frequently called OPC. See *OPC.*

OPC (OLE-PC; object linking and embedding for process control). The task force of a consortium created to rapidly develop an OLE-based interface standard for manufacturing automation to improve interoperability among industrial devices and software. Includes AEG Schneider, Applied Automation, Aspen Technology, Fisher-Rosemount, Fluke, Gensym, Hewlett-Packard, Honeywell, Hitachi, Intellution, Intuitive Technology, Moore Products, National Instruments, Rockwell Software, Siemens, USDATA, and Wonderware.

*See table comparing COM with CORBA.

Being an OPC server in a distributed control system (DCS) is only part of the communication issue. The OPC client needs to use these interfaces to push or pull data from the running system. OPC is commonly used in human machine interfaces, alarm managers, and historians. In the late 1990s the question arose about how fast OPC really is and if it can be used across a *network* connection. While OPC is adequate for communicating with servers running on the same machine as the client, most people like to distribute OPC servers in their own PCs. When an OPC server is located on a separate machine than the client, DCOM needs to be used to transfer data. If the communication transactions are based upon standard marshaling, then OPC servers tend to be slow. There are those at the end of the 1990s who see great promise in this technology but at the same time felt that if a DCS is using OPC for client and server communication for all real-time data, then significant performance considerations arise as the size of the system grows.

Competing Objects to Link Applications

If you began reading the definitions in the sidebar-box, you may have picked up on two different software object oriented programming standards. One uses Common Object

Models (COM) to link applications, the other uses the Common Object Request Broker Architecture (CORBA). It is, of course, a difference between Microsoft®, who uses the former, and the Object Management Group (OMG), a consortium of companies who use the latter. The table compares these two approaches:

Comparing COM with CORBA—COM only rules wherever Microsoft rules, and is extremely platform restricted. CORBA, however, is really the only platform-independent object standard:

COM	CORBA
Advantages	
Supports scripting languages, which makes for simpler development	Platform independent
Native integration with Windows environment	Fast developing synergy between CORBA and Java standards
Has an object component container model	Supports multiple instances
Free with Windows and Microsoft products	
Disadvantages	
Extremely platform restricted (Windows only)	Requires sophisticated programming skills
Outside Windows environment COM is as complex to use as CORBA	Little support for scripting languages
Does not support multiple instance	Lacks a component container model
	Not free, CORBA ORBs must be purchased from a vendor

Communication from Afar

There are times where it is convenient to view the process from somewhere away from the plant site. The blue-sky people could say these could be when salesmen need to view the progress of the order or a manager needs to look into the health of the operation. More realistically such remote process viewing is for the plant engineer troubleshooting control configurations or for a vendor who has permission to look in on some new innovation, or perhaps review a particularly obstinate service problem (they do happen, you know, even in the best of systems).

> Never do you ever want to *control* blindly from afar, except for some very simple actions, like turning a specific piece of equipment on or off!

The standard workstations and networks used in some control systems have made applications such as remote dial-in a simple solution. By connecting a modem to one of the communication ports, you can access the controller data from anywhere in the world that has a telephone (Figure 25-10). From a remote location, you can display and manipulate any program that could be displayed and manipulated at the plant site: operator interface, controller configuration, and the like. Dial-in from a remote site provides benefits such as off-site process and system troubleshooting and allows applications such as unmanned sites. This architecture can be accomplished in a few ways. Dial-in access can be password-protected.

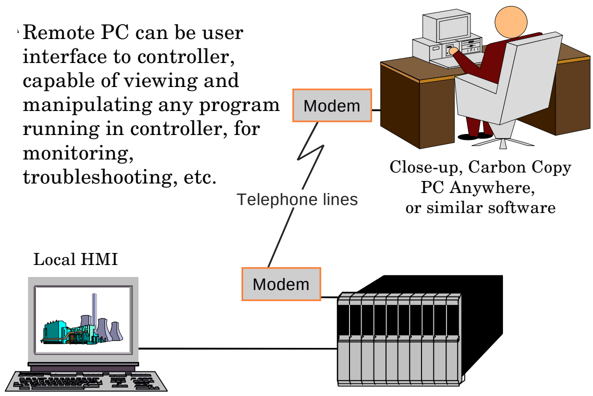


Figure 25-10. Reconfiguration from Distant Engineering Location

In the PC architecture, dial-in can come from a remote PC to the operator interface PC using one of several commercial software packages (Figure 25-11). The monitor, keyboard, and mouse signals are being sent over the phone lines through phone modems at each end. In this architecture, the host PC has the necessary software for its normal function along with the remote software package, while the remote PC only needs the remote software. The software mimics the host PC on the remote PC, allowing the remote user to have full control of the host PC while the user of the host PC watches and vice versa. In some systems, you only need to purchase controller software for the plant site; the remote computer actually manipulates the software that resides in the controller access port.

The second method uses a communications protocol called SLIP (Serial Line Internet Protocol). In the SLIP architecture the appropriate software must be on the remote PC, and only data is transferred over the phone lines. In this architecture, the local modem is connected to the serial port of the controller. In the workstation architecture a remote workstation or X terminal can access a workstation over the phone lines using the X windows protocol and PPP (Point-to-Point Protocol). This option gives the remote station all the functions that are available on a local X terminal.

Spread Spectrum Radio Technology

There are many emerging needs for wireless communications for linking the remote areas of an operation. Off shore oilrigs, pipelines, and even large water plants have this

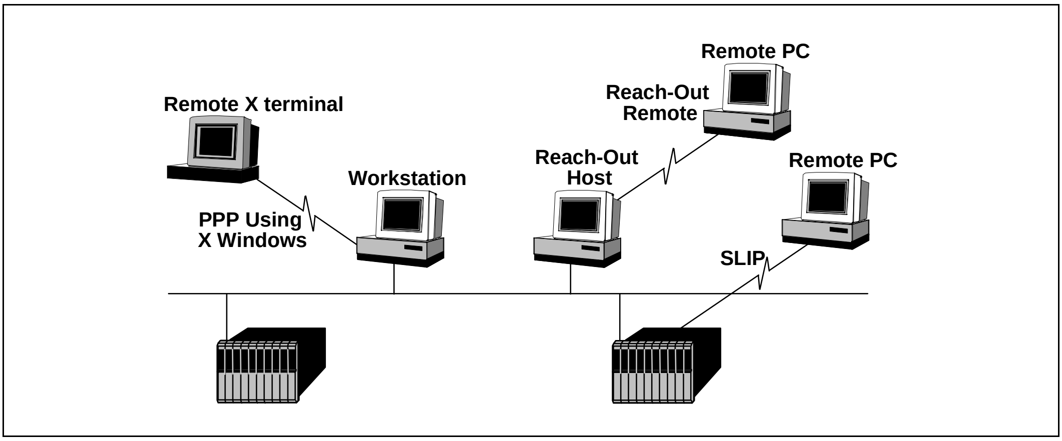


Figure 25-11. Remote Dial-in for Monitoring and Troubleshooting

need. There can be severe limitations to this approach, so it is better to operate as a "true" SCADA system. (As we said earlier, many people erroneously refer to a simple data acquisition and control (DAC) system as a SCADA system). In such a system, there must be very carefully thought out control strategies to allow for frequent, sudden interruptions. "Check-before-execute" routines must be thoroughly understood and implemented.

Several methods of communication can be used, including microwave, satellite, and radio. One radio technology to reduce the impact of interference is spread spectrum signals (Figure 25-12). The spread spectrum signal is created by modulating the radio frequency signal with a spreading sequence code or by "hopping" the frequency of the carrier signal. This was originally developed for the U.S. military to prevent jamming.

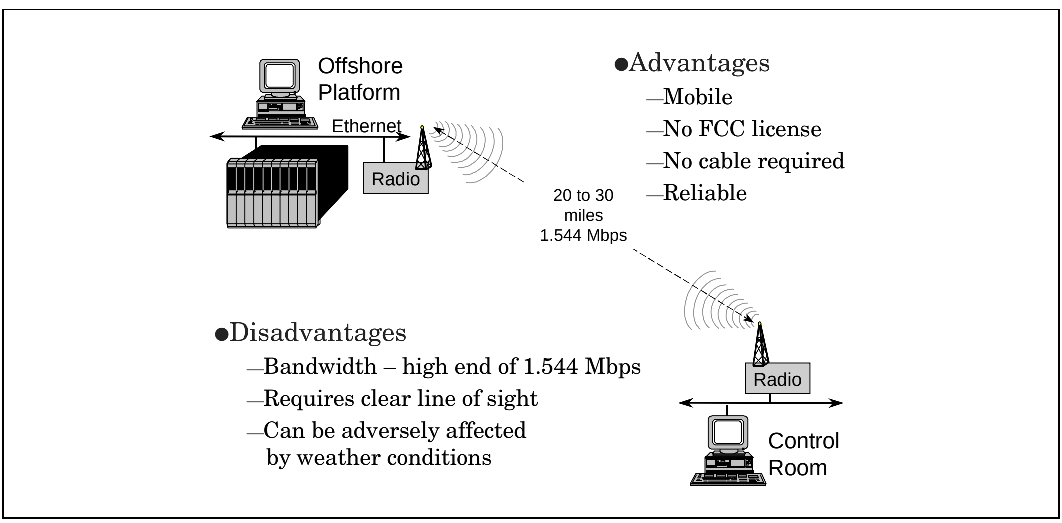


Figure 25-12. Spread Spectrum Radio

Cellular Phone Technology

An emerging wireless digital communications technology in the late 1990s that is useful for “true” SCADA systems is Cellular Digital Packet Data (CDPD). This uses a technique that sends packets of data using existing cellular communications technology for a given region. While cost is a constraining factor with CDPD, this technology has proved effective for regions without direct telephone or leased lines.

Digital Field Communications 26

In the previous four chapters, we discussed systemwide networks. There are also digital networks between controllers and their inputs and outputs. Several "standard" digital networks have been devised, many of which have been encouraged by the vendors who designed them and who hoped for de facto acceptance. There is nothing wrong with this multiple approach, because these designs have been created to fill product gaps left by the lack of any standards to meet specific communication needs!

All of the various network designs have serious merit for their intended use, and many have been innovative enough to be expanded to more comprehensive uses. The one we have selected for discussion here is only one of those that was *not* selected; however, it was an ISA development and this book is published by ISA. Fieldbus Foundation (FF) fieldbus has been picked because it is the first one that has addressed the user layer and, as such, represents the most comprehensive such design to date.

Fieldbus—A digital, serial, multidrop, two-way communication path among industrial field equipment such as sensors, actuators, controllers, and even control room devices. Fieldbus is a specific ISA SP50 (Fieldbus Foundation) standard for digital communications operating at the lowest level of data communications (I/Os) in automation systems. It allows for communication and interoperability among "smart" field devices and control system devices from multiple vendors; it also supports information access for monitoring, control, and alarm tasks during plant start-up, operation, and maintenance. As this standard is developing and gaining interest, two versions are emerging: H1 for linking sensors and actuators to control devices, and H2 for functioning as a full blown data highway on a more sophisticated scale.

Fieldbus Communication Services (FCS)—In the context of the Fieldbus Foundation, FCS is a messaging sublayer that the application layer provides to access remote application process objects and their object directory descriptions.

Why Digital Field Communications?

The problem is that we have had microprocessor-based transmitters, instruments and actuators, using digital technology for years now, but up through the 1990s we are still communicating using an analog technology for communication (Figure 26-1). This limits the information that can already come into the control system as well as the information that can go to the various field devices!

The conventional signal links a transmitter to a device that requires the signal. Typically, this is done using an analog representation of the primary value (Figure 26-2).

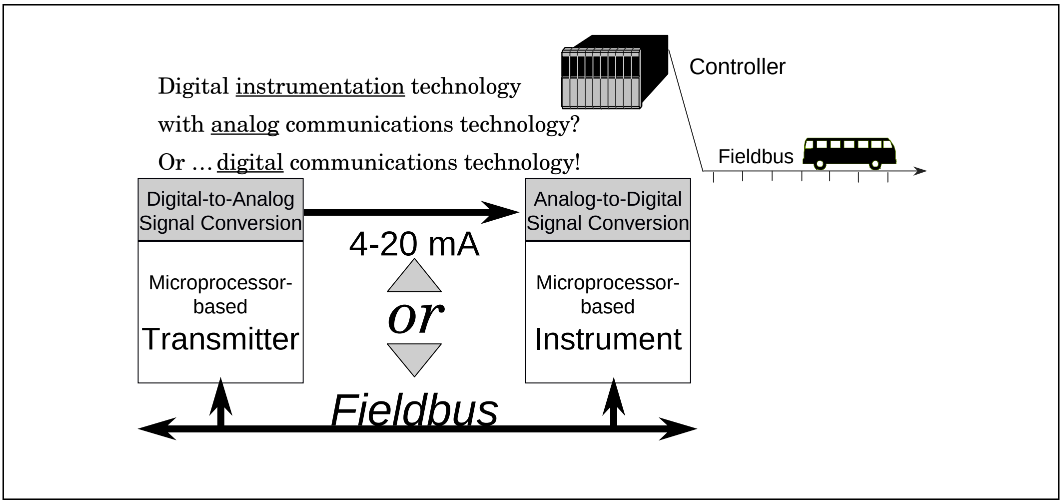


Figure 26-1. Which Is Better. . .Analog or Digital Field Communications?

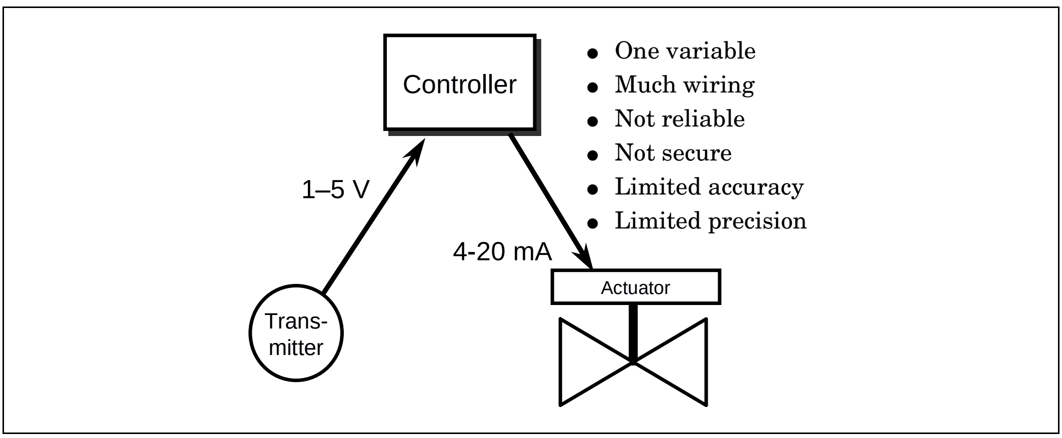


Figure 26-2. Traditional Analog Technology

For actuators, the process is the same. These shortcomings limit the very advantages offered by the remainder of the digital control system.

The basic problem is that microprocessor-based devices are converting their digital values to analog signals, transmitting them and then converting them back to digital signals (Figure 26-3). What is needed is a standard digital communications protocol that allows intelligent devices to share much information and permits a multidrop instrumentation network! (Figure 26-4)

To better understand the evolution of fieldbus, we should first look at the evolution of control and monitoring. In the beginning, there was LOCAL control and monitoring (Figure 26-5).

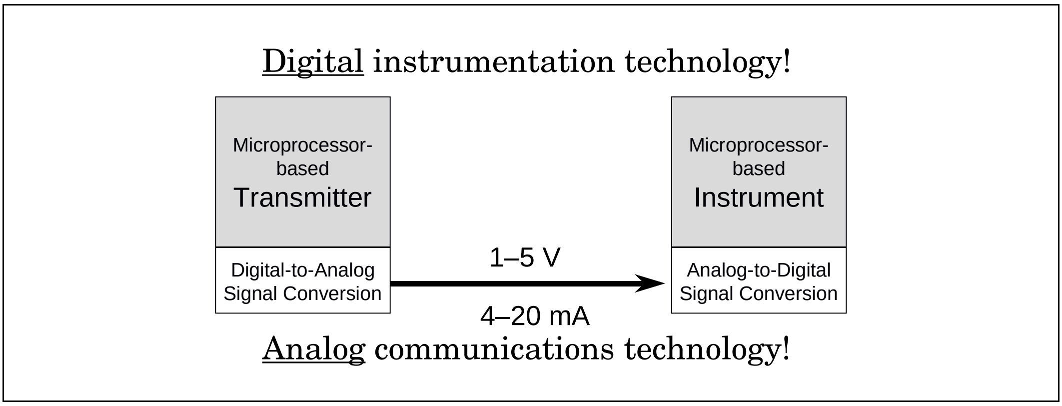


Figure 26-3. Much Wasteful Conversion Exists Today with Conventional Signals

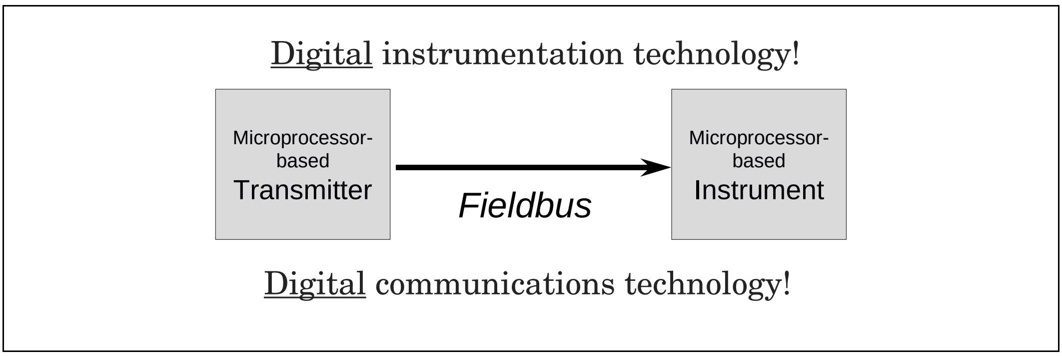


Figure 26-4. Fieldbus Is Specifically Designed For Digital Communication Among Digital Instrumentation

There were advantages:

- Not much wiring

And also disadvantages:

- Not much control
- Not much monitoring
- Not much alarming.

Then, with the advent of the minicomputer the local equipment was replaced with computer equipment at the control room, and all the points were wired to the control room. We called it centralized control and monitoring (Figure 26-6).

This increased the advantages:

- Central view of process
- Flexible control
- Flexible alarming
- History, events

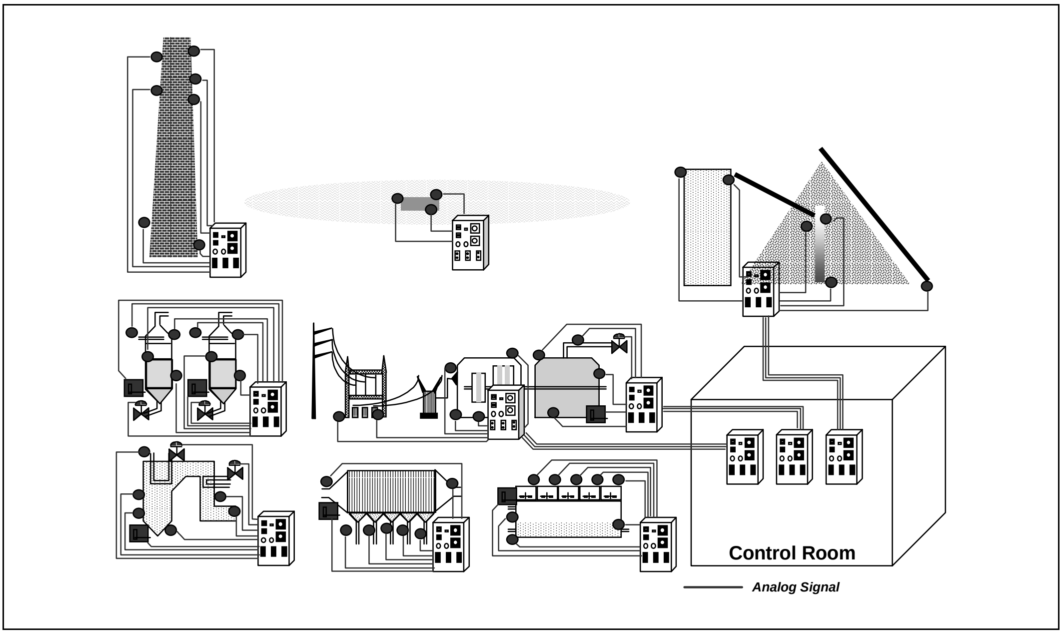


Figure 26-5. When Control and Monitoring Were Local and Analog; Not All Control Panels Were in the Control Room

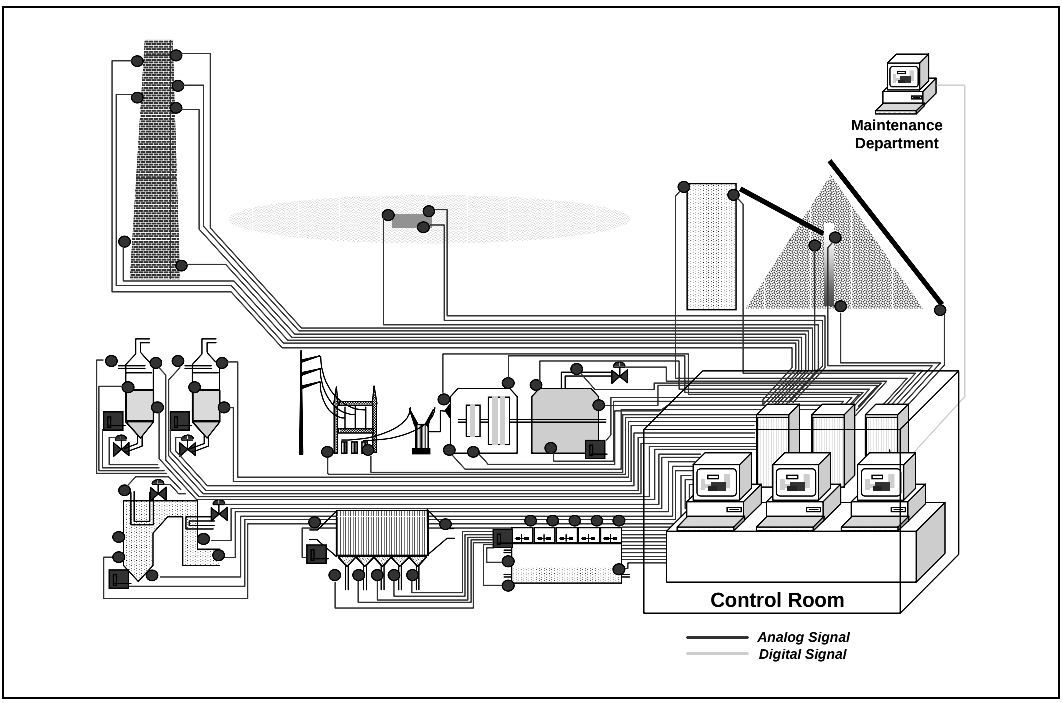


Figure 26-6. Direct Digital Control (DDC) and Monitoring in Mainframe Computer Brought About Central Control From a Single Control Room

But still left with disadvantages:

- Wiring costs
- Risk of losing all control
- Not scalable.

Next, with the advent of the microprocessor the massive wiring was no longer necessary, the I/O equipment was distributed throughout the plant, proprietary digital data highways were added to interconnect the distributed processing units, and we called it DISTRIBUTED control and monitoring (Figure 26-7).

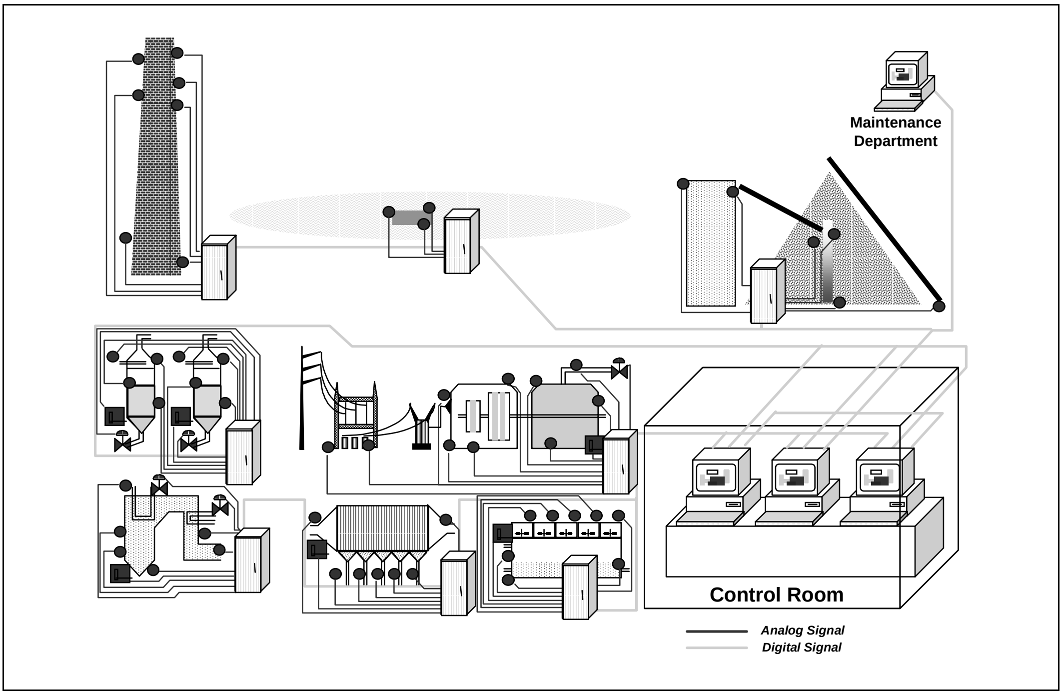


Figure 26-7. Microprocessor Allowed the Design of Distributed Control and Monitoring to Happen

This added to the central computer's advantages:

- Shorter wire runs
- More scalability
- Far less risk of losing all of the process or plant through a simple computer failure.

However, it still retained some disadvantages:

- Still had significant wiring, and
- It could only be made possible through proprietary systems, causing vendor interconnection problems.

Then, however, came a two-way, multidrop, digital communications standard *designed* for control and monitoring rather than modifying a design from another purpose. This approach permitted digital communications lines directly to the transmitters and actuators and allowed the devices to fully exploit their digital capabilities. It was called "FIELDBUS control and monitoring" (Figure 26-8).

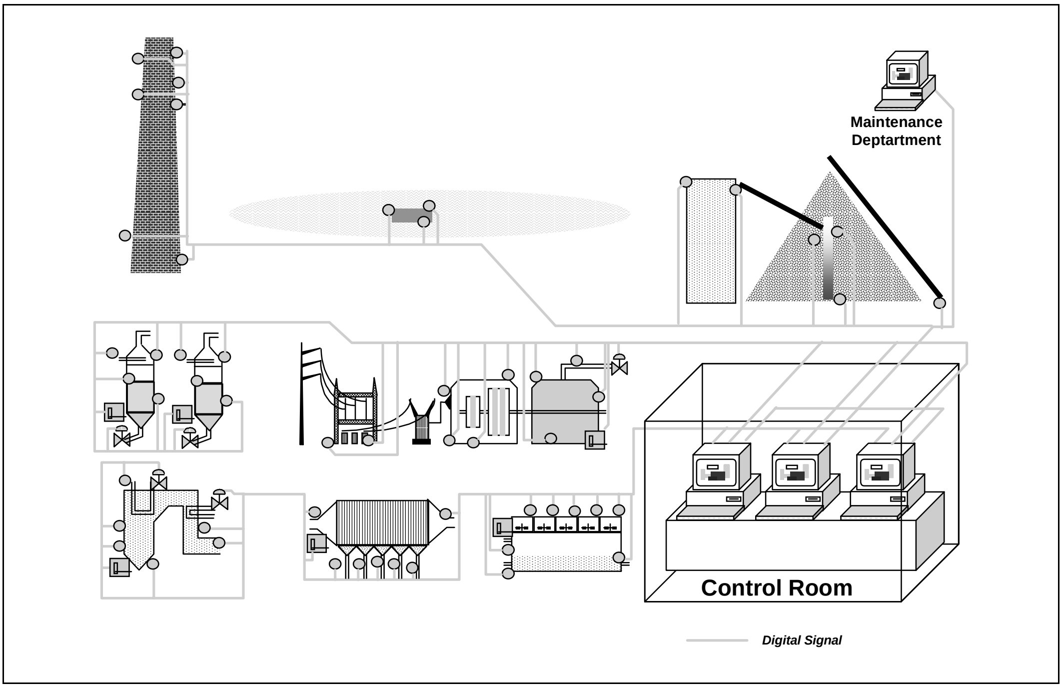


Figure 26-8. Fieldbus Control and Monitoring Significantly Reduces Field Wiring

Fieldbus has expanded the advantages of distributed control with the following features:

- Greatly reduced *wiring*
- Even more *scalability*
- Still less *risk*
- Multivendor *interoperability* (due to an open global standard)
- Can interconnect *DCSs and PLCs*
- Control can be *local* (again!)
- *New opportunities* for advanced features for the manufacturers and the users (e.g., self-tune, fuzzy logic, neural networks).

Advantages of Fieldbus

These architectural advantages to Fieldbus (Figure 26-9), lead us to a list of the benefits of fieldbus which can be placed into three categories: installation, operation, and maintenance.

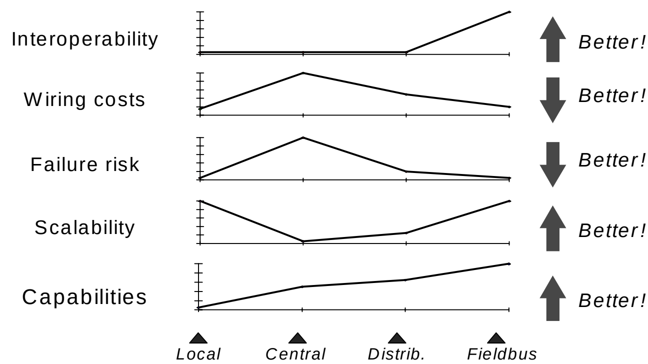


Figure 26-9. Architectural Changes in Control Systems Changed Benefits Picture

Installation benefits include:

- Fieldbus is multidrop, which reduces the amount of wiring and the costs, not just of the wire, but also of the terminations and testing of all this wiring.
- Fieldbus provides standardized access methods to the device parameters of both sensors and end elements over the wire, thereby enabling remote configuration. This improves the accessibility of those devices, especially those mounted in outside environments. The use of digital signals will also increase the accuracy of calibration. Incorrect model numbers, for example, which can "report back" when digitally accessed, will show up before start-ups, saving considerable time and expense.
- Fieldbus defines enough user services to provide interoperability so that users *may select* from multiple suppliers. A single supplier cannot supply *all* of a user's needs! Why should the user have to compromise? (Shouldn't the user be able to take advantage of specialized products?)

Operating benefits include:

- The use of digital floating-point representation permits the transmission of numeric information with no chance of the degradation that can happen with an analog signal. Inaccuracies are not introduced in the transmission step.
- That same seven-digit digital floating-point representation also provides far more precision than its analog counterpart.

- Digital signals eliminate the worry over bad control actions as a result of bad signals. Digital transmission ensures better control, which leads to less wasted product and energy both of which directly improve profit.
- More information will be made available from fieldbus devices because it is so easy to gather information at essentially no extra cost. For example, a transmitter that has temperature compensation for its pH or conductivity calculation will be able to transmit the temperature in addition to the pH or conductivity.
- Digital signals are more secure because they have the safeguards needed to detect errors and degradation of signal due for any reason; whereas analog signals have little or no safeguards. Reliability reduces downtime.

Maintenance benefits include:

- Less maintenance because of the increased reliability of digital technology.
- Faster maintenance because digital diagnostics can be specific, leading to faster correction and complete, automatic documentation.
- Access to numerous parameters within smart devices makes *remote* diagnostics possible (Figure 26-10) and sometimes maintenance as well.
- An open standard of this depth permits the *interchangeability* of similarly functioning products. Therefore, at replacement time the user again can select the most appropriate vendor.

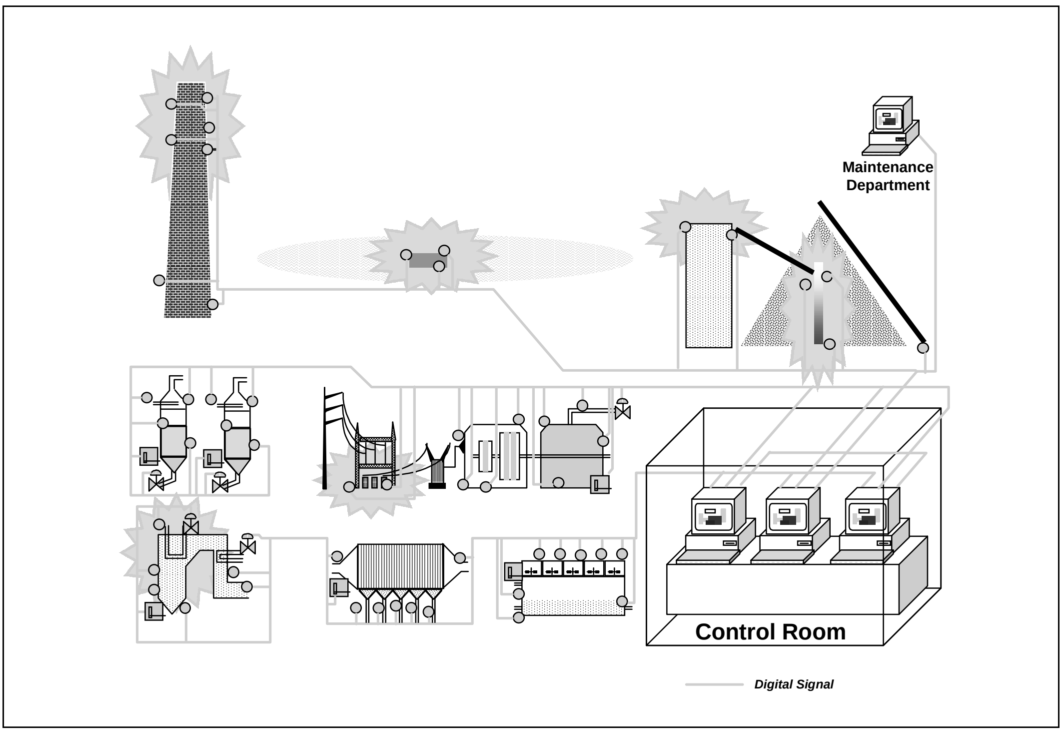


Figure 26-10. Fieldbus Allows Remote Diagnostics Through Standard Access to Calibration and Maintenance Attributes

Consider those devices that are high, hot, wet, located in dangerous places, and hard to access. Fieldbus permits standardized access to calibration and maintenance attributes as well as to the information provided by the vendor. A disk from the vendor can be used to load appropriate information into the user's workstation to allow access to any of the features provided by that vendor, including calibration and diagnostic information.

Interoperability and Interchangeability

What do these two terms mean? Interchangeability permits you to replace a device from one vender with that of a similarly functioning product from another. They will each have the same complete access to other devices on the same network. For example, no "reprogramming" of other devices by the user will be necessary to accept the change.

Interoperability, on the other hand, is the ability to tightly interconnect devices from two different vendors. They will operate together both properly and closely, sharing status and all parameters.

Fieldbus provides interoperability by defining the following:

- Electrical signal
- Media access protocol
- Communications handshaking protocol
- Supported data types
- The method to describe the device over the wire
- Comprehensive function blocks
- Modes and status
- Cascade initialization, fail-safe propagation
- The alarm and event reporting mechanism

Differentiating Fieldbus Vendors

A common fear regarding standards that are detailed enough to ensure interoperability is that they will prohibit innovation by holding vendors to the functions in the standard. However, vendors have plenty of room to show product differentiation and innovation:

- Quality of sales, training, delivery, documentation, service, and support
- Quality of the product, accuracy, precision, and robustness, maintainability
- Superior measurement/actuator technology
- Functions beyond the standard
- Application expertise, both through added features and assistance to user
- Value, ruggedness, features included

None of these are addressed by the standard.

How Fieldbus Works

Fieldbus uses a subset of the ISO's OSI Reference called Enhanced Performance Architecture (EPA) model which omits Layers 3 through 6:

3: Network 4: Transport 5: Session 6: Presentation

Fieldbus uses the other three layers (Figure 26-11):

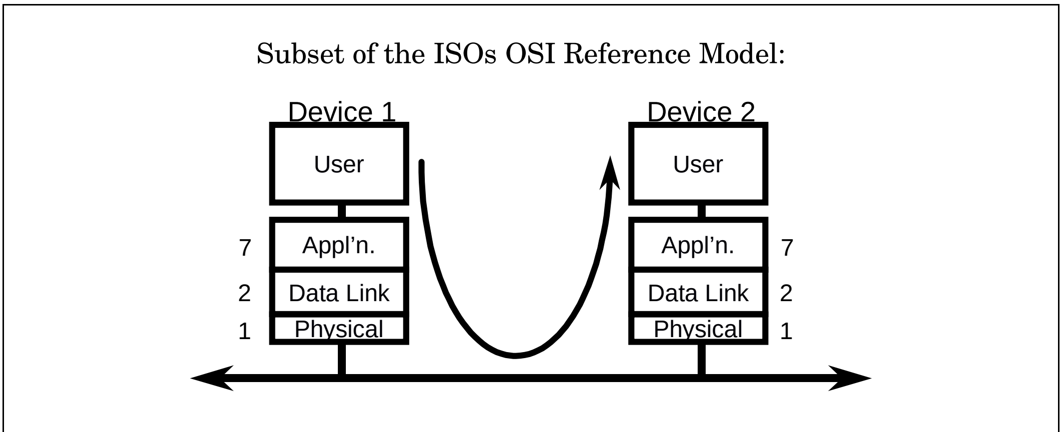


Figure 26-11. Fieldbus Uses the Enhanced Performance Architecture (EPA) Model

1: Physical Layer

—Signal characteristics, preamble/postamble, frame check sequence

2: Data Link Layer

—Media access protocol, reliable message transfer, cyclic and acyclic services

7: Application Layer

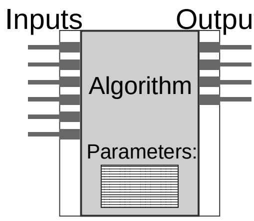
—Naming and addressing, variable access, uploading and downloading

In addition, Fieldbus uniquely has included a user layer. This layer defines the function blocks with mode and status, events and alarms, and device descriptions (Figure 26-12). The user layer is device oriented and database oriented and is not considered a communication layer. The application layer that is the highest layer is message oriented.

Because the user layer defines the behavior of the device, it is the most important layer.

A **function block** is:

- An algorithm
- Set of defined inputs, user-connected
- Set of defined outputs, user-connectable



- Analog input
- Analog output
- Discrete input
- Discrete output
- Pulse input
- PID
- Step PID
- Manual loader
- Ratio station
- Bias/gain station
- Control selector
- Lead/lag
- Dead time
- Set point generator
- Signal characterizer
- Signal selector
- Device controller
- Timer/counter
- Arithmetic
- Calculator
- Integrator
- Complex analog out
- Complex discrete out
- Analog alarm
- Discrete alarm
- Analog human I/F
- Discrete human I/F

Figure 26-12. Fieldbus User Layer Defines Function Blocks for Many Elements of Control

- Set of attributes:
  - — Limits
  - — Tuning Parameters
  - — Constants
  - — Miscellaneous specifications and parameters

The analog input function block (Figure 26-13) includes process variable and scale, signal output scale, linearization, alarm limits, and alarm priorities.

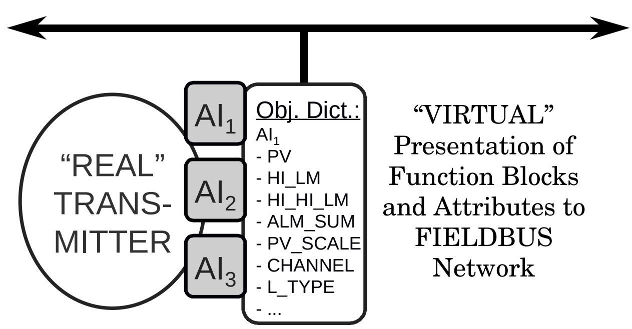


Figure 26-13. Analog Input Function Block is Like a Virtual Fieldbus Device

Virtual three-mode (PID) controllers on the fieldbus (Figure 26-14) network through the object dictionary give access to tuning constants, gain, reset, rate, feedforward gain, mode, alarm limits, description, and units of measure.

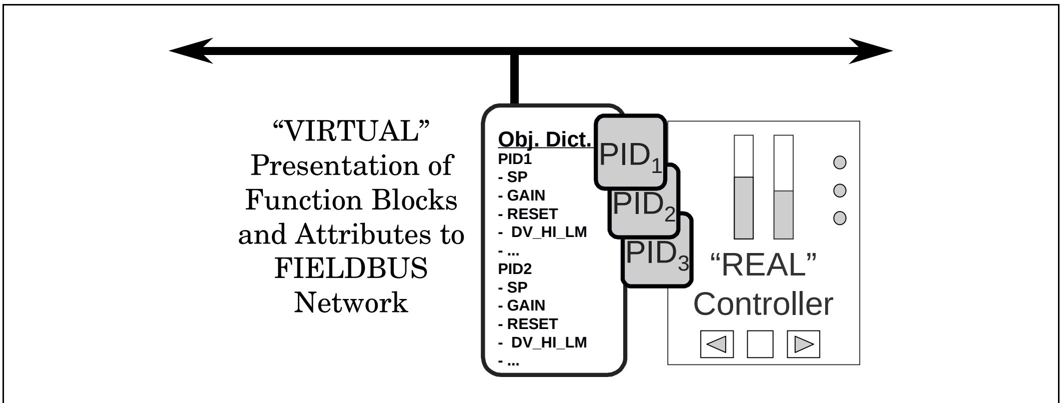


Figure 26-14. Fieldbus Object Directory May Present One or More PID Controllers

An end element such as a valve (Figure 26-15) can also be presented to the fieldbus network through the object dictionary and includes cascade input, output range limits and units, as well as fail-safe condition and action.

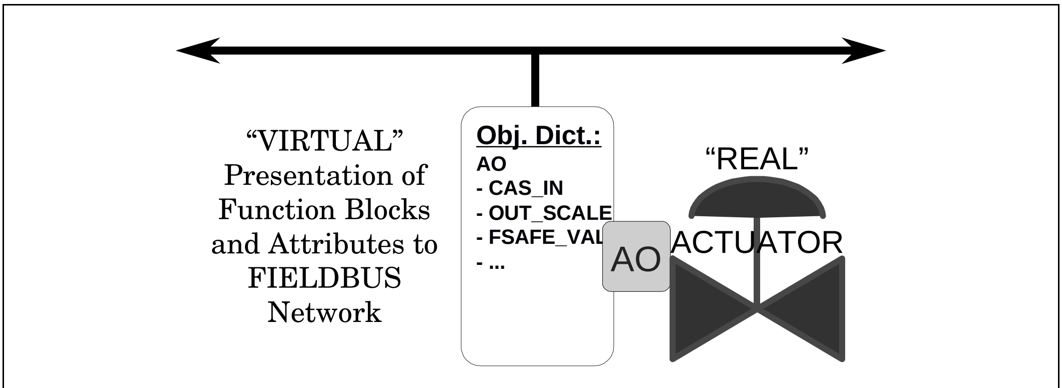


Figure 26-15. End Elements Are Also Presented to the Network

Today three-mode (PID) controllers may be found locally in any input or output device, such as a valve or transmitter. The Fieldbus user layer has already been designed to accommodate these and especially to allow these "field controllers" (Figure 26-16) to couple tightly to other parts of control loops located in products from other vendors. This will dramatically alter the possibilities for using process control strategies.

A control loop can be configured directly over the fieldbus (Figure 26-17). For example, the connection between the analog inputs and the three-mode (PID) controller is provided with automatic cyclic updates of their values, including the status of the input variable.

Typical fieldbus devices today include transmitters, actuators, controllers, indicators, and recorders. Consider, however, the impact Fieldbus will have on the use of handheld

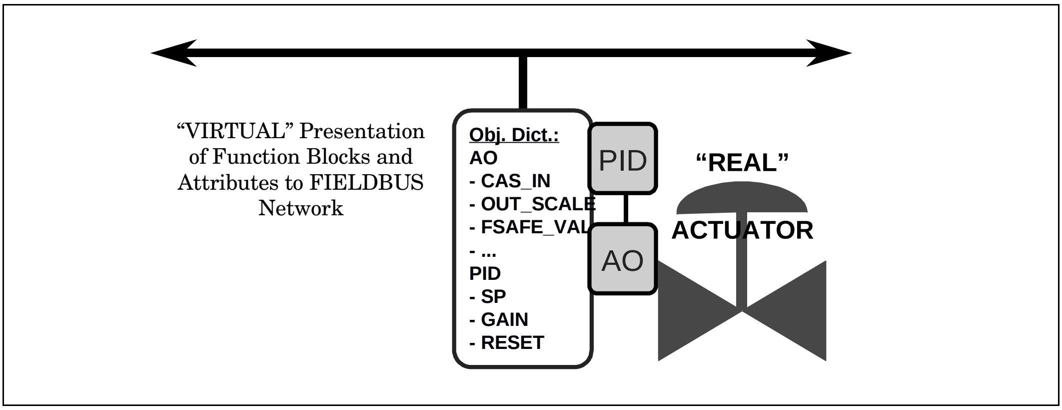


Figure 26-16. Fieldbus Allows for "Smart" Transmitters and End Elements

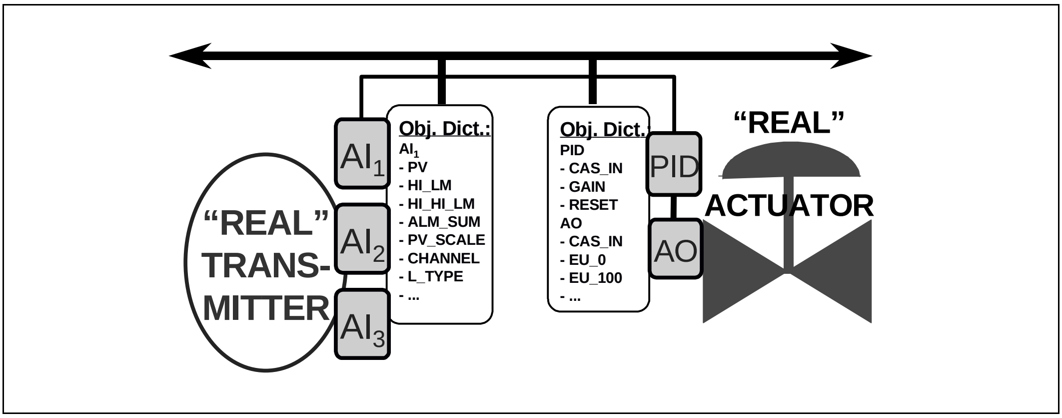


Figure 26-17. Control Loops Can Be "Spread Out" over Fieldbus

devices, local graphical user interfaces (GUIs) rather than only central control rooms, the role of PCs, and the entire architectures of distributed control systems (DCSs) and programmable logic systems (PLSs).

The Fieldbus Foundation

The fieldbus Foundation (FF) is a group of companies cooperating to accelerate a single international fieldbus. They banded together (quite painfully) because the development of the IEC/ISA SP50 standard had been taking too long (well over a decade). They saw the concept of fieldbus as too valuable to ignore but felt that most of the needed technology already existed to accelerate the pace. They also felt that an expedient compromise was necessary to provide desperate users with real products (Figure 26-18). Those compromises, however, have been made in such a way as to support future work.

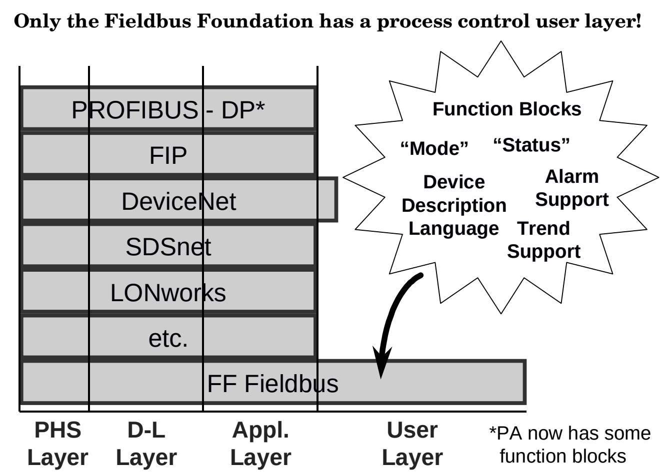


Figure 26-18. Thoroughness, Such As Including the User Layer, Has Challenged Fieldbus Development

The Fieldbus Foundation is unique in that it supports extensions *beyond* the standard. The group supports *VDS (Variable Definition Syntax)*, which is essential to enabling innovation without requiring frequent version upgrades. Viewed as quixotic by some, the Fieldbus Foundation regards this aspect of its effort as its strength!

VDS (Variable Definition Syntax)—This is DDL renamed, with the copyright granted to the SP50 Committee by the Interoperable Systems Project (ISP) Foundation.

DDL—Device Description Language from Highway Addressable Remote Transducer (HART), and now called VDS (not to be confused with data definition language, used in database management systems).

Architectural Issues 27

Rarely does a control system function in isolation from other computer systems in a plant. Several situations call for supplementing the control actions with data from other operations in the plant, gathering data from the control system to provide other operations in the plant, or both. Connecting these, of course, alter the architecture of the system. The nature of the impact depends upon the frequency, quantity, and speed of communication needed for the functions involved.

Impact of Control Strategy on Architecture

A collection of loops, sometimes in separate controllers, used in either a small laboratory or on a unit process that is a stand-alone needs only a simple structure. It may have a simple bus interconnecting each of the stations, and perhaps one of the stations might control the communication and in effect be a central operator interface, such as you would find in a personal computer (PC). This communication link would not need, as fast a response as would be required on larger systems because these small systems rarely require sophisticated multivariable control. Any multivariable control is likely to be limited to a cascade pair of loops, and those probably will be done within a single control station.

A more complex situation is having multivariable control in a larger processing plant. Here, the impact of multiple processors, linked together from different areas of the plant, may very likely control the performance, speed, and/or quality of the other processes within the plant. An example is a complex chemical operation in which batching and continuous operations are blended together within the same plant.

The use of advanced control techniques so far has usually been limited to unit processes. Optimizing a control loop is often a question of modifying the gain and/or reset action of a control loop from some influence outside the loop itself but within the same process unit. If such an automatic function had to come from the other end of the plant within a critical response time, then a very sophisticated network would indeed be required. For managing sophisticated loop optimization a good input/output (I/O) network is generally needed. If the I/O is from a digital network, then it should be very local, perhaps only to that particular controller and very likely a parallel link.

Impact of Advanced Control

More advanced control techniques are now emerging, such as model-based control, as well as more advanced control strategies based upon fuzzy logic, neural networks, generic algorithms, and chaos theory. These technologies generally require a tremendous amount of computation, and in process control these computations must occur essentially in real-time. The practicality of these methods will increase as microprocessors become faster and more powerful and local memory becomes larger, requiring much smaller space.

These advanced strategies, which are often used in combination, are becoming practical in overall plantwide strategies. As a result, advanced control has been relegated to an "external" computer that gathers data from a very large number of variables. In such a role, the advanced strategies are designed for suggesting guidelines or creating long-term adjustments to operations. Nevertheless, this "independent" computer may indeed reside on the same network. As advanced control techniques become faster, communication response times of the differing applications on the network must be matched and not limited by protocols and connections.

Subsets of fuzzy logic and neural networks are now being used within local controllers. They are nearly always a supplement to the use of traditional three-mode (PID) control technology for obtaining functional completeness. As newer technologies are able to migrate into local controllers, the "strain" will also be removed from the communications system. Remember, however, these new technologies are for multivariable strategies, not single loops involving the matching of process variable (PV) with set point (SP). Much depends upon where these variables come from!

Dilemma of Multiple Vendors

There are many kinds of control function blocks, such as PID or three-mode control, sequential control, logic control, motor control, and variations of these. Quite often, there are products that do logic control without very sophisticated process control, other products that do very sophisticated process control and are very skimpy on sequence control, and so on. The degrees of control function block sophistication will vary with each vendor's experience with that type of control.

Motor control, for example, requires the use of many interlocking blocks and overrides and often is relegated to programmable logic controllers (PLCs). Few vendors experienced in process work have a complete function block for this capability built into their product. Yet the PLC vendor may not have a single function block with all the sophisticated features built in. Some PLC vendors do not use a function block approach. This results in the need for the user to create the features upon each use of that function. Even with "cut and paste" techniques, upon reconfiguring, the user may inadvertently omit portions or develop time-consuming procedures governing on how to maintain consistency.

Frequently, when several of these functions are brought together in a common system, such as motor control and process control, there is a need to link different products together. Interlinking controller blocks between different vendors but on the same network presents problems in the way they communicate between each other. If these blocks are separate devices, they're quite often required to have peer-to-peer communication, meaning that one box can talk directly to one of the others on a common network. Often this becomes some direct "external" link to one of the stations as a subnetwork.

Connection to External Hardware

Many systems allow external devices to be connected directly onto the network. Although these will generally communicate using the same protocol, this is not sufficient to ensure that no loss of performance is incurred. Because many external devices have different data processing requirements and different database update rates, they may need to communicate at different rates.

Therefore, a computer connected directly to the network may tie up a disproportionate amount of bandwidth for communication that has nothing to do with the direct process control application. In a similar way, the device performing high-speed logic required in detection of sequence-of-events may need to off-load large amounts of data in a relatively short time period. In such systems, the basic performance of the control system suffers, and in some cases the data highway or communication network can stall as a result of the integration of these peripheral devices.

One solution is to provide a separate subnetwork for these external devices, but this has two main drawbacks. The first is that the incorporation of these devices under a separate network means that true integration into the control network database is difficult, and many of the advantages of the so-called open architecture are lost. The second is that adding separate hardware and additional communication software modules for the networks adds to the complexity of the system and thus reduces its overall availability by adding disproportionately to the total component count.

In Figure 27-1(A), the computer is connected directly to the data highway, perhaps for control optimization. In Figure 27-1(B), the connection to the computer is through the operator's station, such as would happen for gathering history. In either case, the external device links should never impede the control performance!

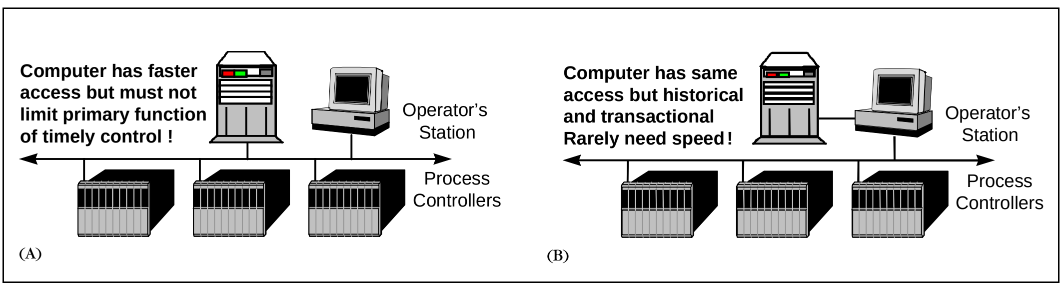


Figure 27-1. External Device Links Must Not Impede Process Performance!

There are systems offering "open architecture" that are combinations of totally different hardware communicating through multiple layers of networks, gateways, repeaters, and the like and yet purporting to be integrated systems. Do not confuse integration with interfacing.

Interfacing External Devices—More Nuts and Bolts

Business structure is significantly different than plant control structure, and as a result the two require different database types, different memory capacities, and different processing speeds. Therefore, they need different kinds of computers with different kinds of databases. Although these machines may talk to each other, they certainly make it very difficult to have one common computer with both navigational and transactional activities within it. As a result, there needs to be an interface between these two kinds of computer systems. This is no different than the links between any two different computing devices that were not specifically designed to communicate with each other. The extent of the interface depends upon how different the devices are.

When number-one son was in kindergarten, he asked one night to have a “Show and Tell” project for the next day. Not to be outdone by other parents, I showed him the old cup-and-string telephone (By the end of the week that school was awash with cups and strings. They hated me there for months). Excitedly, he took it in and demonstrated with a friend as I had showed him. At first, both kids placed their cups to their ears to listen. Nothing happened, of course. Onlookers were confused.

It was just like two people discovering they both have an RS-232 port on their computers. “Hey,” says one, “let’s plug them into each other, and they’ll work together!” Surprise! They don’t. They just sit there and listen and listen. Of course, our kids realized what was happening, so they both turned and began to talk. Realizing that didn’t work so well either, they devised a signal to notify each other when to talk and when to listen. Hardly unique, that signal quickly reduced to the word over, which evolved from them saying, “now switch over.” How original. They had discovered a protocol, and *they* were the “driver package.”

When two devices are interconnected, the first requirement is to determine if the driver package resides in either of them or in some separate device connected between them (Figure 27-2). When deciding what external computers or other networks to use, this is one of the requirements. Others include what kind of data (information) must pass between, how much of it, how frequently, and in what form. Then it must be decided who is responsible for programming that link, the user’s engineers, the system vendor’s engineers, or some third-party system integrator. What is involved? Well, here comes another analogy.

Picture if you will a railroad system with a main line stretching from New York to Chicago. Along this route, there are many kinds of industries, let’s say a mining operation, a logging operation, and some steel mills. Now on this main line between New York and Chicago, there are standards to be met, such as the width between the rails. Various railway cars from many different railroads, from Canada down through Mexico, must be able to operate over the same lines. Among the standards to be determined are the kinds of couplers between the cars, standards to the dimensions of the cars for handling the weight across the bridges, the dimensions within tunnels, and so on. All these standards would be made to operate according to the normal function of transporting goods and people from the one point to the other.

Now, one of the industries along the line is a mine. It would be very impractical to take an 85-foot-long passenger car and stuff it down a hole on the line to get the miners to the mine face, or for that matter to take large vehicles and try to haul the ore or coal

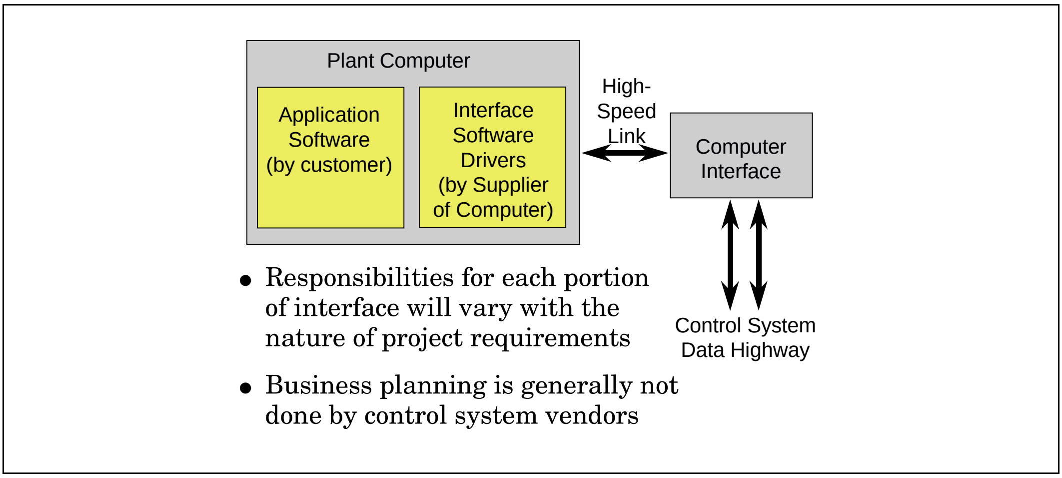


Figure 27-2. Every Computer Interface Needs a "Driver Package"

from the mine face back to the surface. As a result, specialized vehicles to carry people or the coal or ore mined would be designed to fit the space and relevant requirements.

Another industry on our line is a lumbering operation, which requires that you have lightweight rail with the rail closer together than is standard, usually two feet apart. This is so the track can easily be relocated as the trees in a given area are cut. The operation also needs a different-shaped car that can easily navigate the very rough terrain, quickly laid track, and narrow winding curves, and yet support those heavy logs.

The same thing occurs with the needs of a steel mill, another customer on this line. There, different shapes of vehicles are needed to carry molten steel around, something you don't deliver over the main line. The owner will want to move product throughout the plant according to the schedule of the steel mill's production, not the schedule of the main-line railroad.

So whether it be for the speed, or flexibility, the timetable or capacity, or the track dimensions, each of these customers requires different standards than the main-line railroad. The same is true in computers. Different kinds of computers used for different functions take on different dimensions of capacity and speed. And all dimensions capacities and speeds do not fit every circumstance.

There remains a need; however, to connect all these efforts so they can work together. In the case of the railroads, you have railway depots that are between the two systems. The size of this station or depot depends on the varying numbers of people or goods that need to be transported from one system to the other. Everything must be unloaded from one car with one set of dimensions and repackaged to be loaded on a car with different dimensions. The frequency of connections will determine how big the depot has to be to provide service for the different railroads. For example, the size of the depot will increase as the amount of time people or goods wait at the station increases.

So also in every computer interface: there must be sufficient capacity, and processing speed, and the communications link on either side must match. The one side must be sized for the amount of data it carries and the speed at which it arrives, and the other side must match its capacity, speed and packaging requirements. This brings us to the various ways in which computers communicate with each other.

There are some occasions where the computer must make a single query to the other side. In a distributed control system (DCS), for example, the request is for a calculated piece of data out of the computer for a controller. In the case of the computer, the request is for some parameter or temperature, pressure or flow coming from the DCS. This single query must go to the "station" and wait for the next train along the main line, then get off at the correct stop for the data. Having the answer, it must wait again for the correct train to take that data back to the originator.

Although a single query may be necessary on occasion, if there is much data it is very inefficient. Regular "pickups" and deliveries can be arranged to gather larger collections of similar data. The railroad would typically "spot" several cars to be loaded at the customer's site as product becomes available. Then the cars would be gathered together and moved all at the same time, using the same locomotive (communications overhead). In the same way, data can be transported in large blocks of preconfigured data, which would be transported as a single package; when possible, it is far more efficient to move many blocks of data with same overhead (same locomotive) than to move each block individually (one locomotive for each railway car)

The railroad would call this a "unit train."

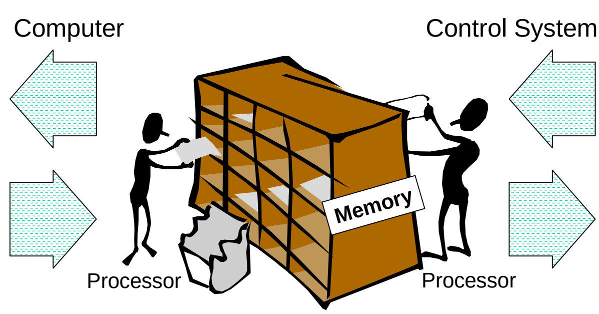


Figure 27-3. Connecting Computers through "Mailboxes"

There is another method for connecting known as a *mailbox*, used for a variety of messages coming in and being sorted at the depot into different categories to be picked up by the "other side" for the specific requests or answers being searched (Figure 27-3). In this scenario, enough memory would be needed for the appropriate number of mailboxes and their respective sizes to allow for delivery cycle. Traffic will come from the computer for inquiries made of the control system and vice versa. Answers would be located in an appropriate spot within the depot for pick-up during the normal delivery cycle used by the control system and/or the computer. The role of the computer interface would be to manage the traffic flow, whether it is of large group, single queries, or the "mailboxing" function that occurs between the two disparate systems.

Integrating External Hardware into the Data Base

The best solution for incorporating external hardware is to provide clearly defined methods and locations for the connection and to supply a data base location specifically for the purpose of making that connection. When other process-level devices are connected, such as programmable controllers, analyzers, data acquisition systems, single loop controllers, and the like, this link is achieved directly at that process level using the control station as a gateway. Every control station can be equipped with a serial interface and a relevant protocol for the external device can to installed within it.

The communications to the external device is handled by a dedicated microprocessor within that control station, so no other system functions will slow it down as a result of the connection. The data received by an external device is placed directly into this control station database, and can be used directly in graphics, trending, reports, and calculations. In a similar way, the data to be written to the external device also comes from the control station's own database, which has its own collection of process variables and set points, and the data may come from a calculation, a control action, an operator input, or an applications program running at any level.

It is therefore possible to connect external process-level devices to the control system network to provide a completely integrated database without degrading the system performance.

Links to Integrate with Plantwide System

The operator station link can also have a dedicated serial interface, again supported by its own processor. This link can support standard protocols such as DDCMP, or HDLC, or X3.28, which are commonly supported by various computer manufacturers. This link provides access to the operator station process tags and display database as well as to the historical data being trended on the system. Real-time data is available across the control network from the various control stations to the operator stations, and available to the computers only through those operator stations (Figure 27-4). The important thing is that this control network access by the computer is controlled and limited by the operator station and that it's therefore impossible to overburden the control network with most computer requests for data. Using this technique, it is still possible to obtain several hundred data values a second.

Consequently, the operator station in this network would be the logical conduit for data moving to the plantwide system. By using separate processors for each of the links, there is no speed delay in moving the data about and in particular no speed delay in the operator functions, which of course must never be impeded. Separate processors also provide the plantwide system with all of the information that resides in the operator station itself and the database of the operator station. Generally, on a plantwide system the information gathered within the operator station is essentially the same information as needed by the plant wide system. Rarely does the plant wide system need any information beyond what has already been reported to the operator station.

The link from the operator station to the plantwide system certainly does not need to have the speed. Rather, it needs to be able to handle the volume of data coming from this one operator station along with many other operator stations. This link provides

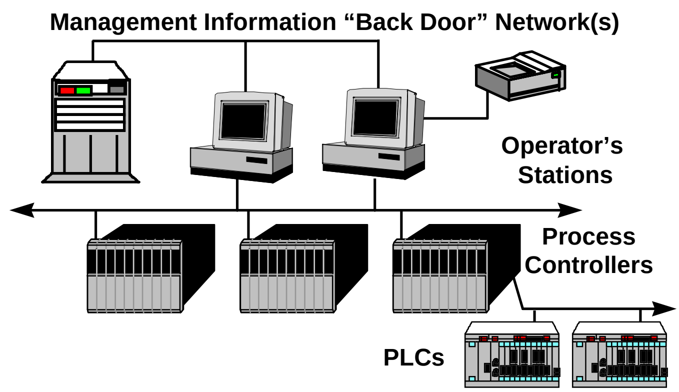


Figure 27-4. "Back Door" Network(s) Reduces Load on Real-time Process Networks

another layer of networking, but its performance and capacity requirements would be far different from those needed for the control system itself. The link also provides a logical division between the two networks. This prevents the network that has the need and desire for information from different parts of the plant from interfering with and delaying the actual control actions of the process itself. A very important buffer is provided here, which is probably a good reason never to allow a "business computer," to be connected directly to the process control network.

The main point to emphasize is that the external devices need access to the system *data* and not to the system itself. By using a dedicated processor to support these links, and by using these processors to allow controlled access to the system's own integrated database, fast, secure access can be granted without degrading the control system's response to standard tasks.

Connecting the Enterprise 28

To be successful in a very competitive marketplace, corporations must think more about making improvements to their entire enterprise. Such thinking must go well beyond the production lines. All corporations must adapt to changing business conditions, improve quality and productivity, strengthen internal controls, and manage costs. Solid business planning is essential to making this all actually happen.

Managing the Enterprise

For a number of years, computers have been used to help implement the planning process. Typically, however, these computers have been disjointed tools used in separate departments and often connected by monthly, quarterly, and even annual reports on paper. Once analyzed, some of the interpreted data may be manually entered into the other computers. Only recently have these business computers been linked together, and true integration of information is just emerging.

ERP—enterprise resource planning; computer based corporate planning system defined by GartnerGroup, linked through Manufacturing Operations Management system to plant controls; counterpart to COMMS by Advanced Manufacturing Research (AMR).

GEMS—global enterprise-wide management system; beyond enterprise resource planning, this includes connectivity from a corporation to the companies of suppliers as well as customers, often over the Internet, to manage the flow of business requirements and products among all of them, as defined by Automation Research Corporation (ARC).

MES—manufacturing execution system; software packages for such functions as plant management, supervisory control and monitoring, plant engineering, and quality management; model concept developed by Advanced Manufacturing Research (AMR), in late 1980s with intention to describe system which, rather than focusing on measurements of material usage or process control, "centers on product itself as it moves through plant on way to customer;" intended to bridge real-time information gap between planning (MIS) and controls (PCS) to link operators and managers with current views of all processing resources; counterpart of MOM model developed by Gartner Group.

MIP—middleware integration platform; provides capability to connect multiple control systems to multiple manufacturing &/or business applications; integration functionality includes event handling, messaging, workflow transactions, systems management, dynamic configuration, some form of database management, etc., as defined by Automation Research Corporation

MOM(S)—manufacturing operations management (system); software packages for such functions as plant management, supervisory control and monitoring, plant engineering, and quality management; model concept developed by Gartner Group, Stamford, CT, USA, intended to bridge real-time information gap between planning (MIS) and controls (PCS) to link operators and managers with current views of all processing resources; counterpart of MES model developed by Advanced Manufacturing Research (AMR).

MIS—management information system; computer network which allows managers to track performance of plant at close to real-time without excessive need for paper based systems; primarily used for planning and scheduling of resources.

OCS—open control system; term used by some for next generation of DCS where ideally hardware and software is not proprietary.

PCS—process control system; responsible for manufacturing of product, as compared to MIS, context being used by vendors of PLC systems (and some market research firms) as equivalent to DCS.

Much talk has been circulating about actually connecting the plant floor to the business operations (Figure 28-1). With enough programming, this can be done, of course. Nevertheless, it requires both scarce time and significant amounts of money.

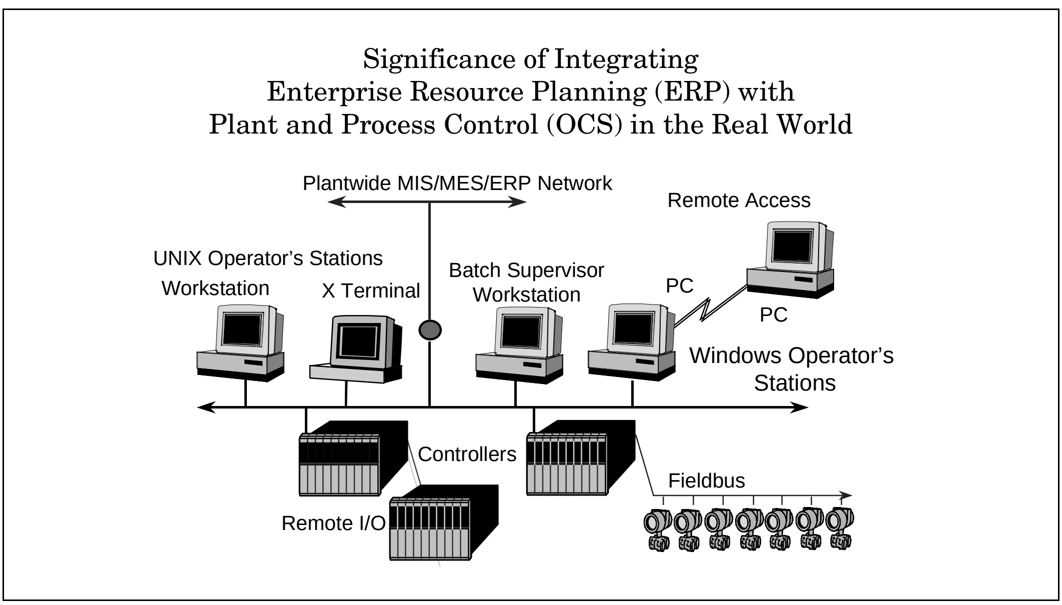


Figure 28-1. Communication Must Exist between the Plant Floor and the Business

Corporations Must Reengineer Continuously to

- Respond to market changes
- Respond to technology changes
- Reduce internal fear of change
- Reduce business perturbations
- Manage costs (margins)

When corporations reengineer as a response to changes in market conditions, it is usually a strategy to regain their market losses. All change is disruptive! It will always initially cause some loss in business. The underlying concept is that the effects of the change will more than recover the losses caused to bring it about. During change, the flow of business is disrupted, causing customer uncertainties. Employees also become uncertain as changes in techniques and skills cause them to lose productivity and experience morale problems.

Ideally, a corporation will make changes proactively rather than reactively. Change should be constant and gradual rather than a traumatic bump:

- Customers fear your change because they may no longer count on the consistent product or service that they have become comfortable with.
- Employees fear your change because they are forced to question their skills and their own abilities to suddenly change their existing ways, which they understand and do well.
- Both of these reactions apply to everyone at all levels, especially the managers, if they are honest with themselves.

All of this creates considerable difficulty for a corporation trying to manage costs and margins!

Managed, Continual Change

- Customers see managed as *positive* because it leads to continual change

— Product growth and improvement
— Service growth and improvement
— Company growth and improvement

- Employees see managed as positive because it leads to continual change
  — Personal growth and improvement

Managed, continual change, is a very positive strategy:

- Customers see this as product, service, and company growth and improvement.
- Employees see such change as leading to growth and improvement for products, service, and the company itself.
- Management also sees such change as an opportunity for growth and improvement.

So we have:

- Sudden traumatic change(s) = fear, uncertainty, initial loss of business
- Continual, managed change = enthusiasm, control of destiny, improvement of business

Corporations require both people and equipment to survive. Change impacts both upgrading and involvement. Equipment will sometimes reach a limit to in its ability to be upgraded. As long as the right tools are used, people reach that limit and usually go well beyond it.

Properly directed change, that uses the appropriate tools and allows on line training without loss of productivity and will improve on-the-job performance. The more gradual the change, the healthier the results, and the traumatic bumps that cause employee unease and business perturbations will be avoided.

Properly Directed Continual Change *without* Traumatic Bumps Needs

- Proper business parameters
- Continually updated information from
  — Process control systems
  — Factory automation systems
- Continual planning
- Acceptance by plant personnel
  — Focus on production
  — Focus on quality

To come closer to achieving this ideal change, proper planning is imperative. Inputs require a continual flow of changing business parameters, blended with a continual flow of current information from the actual operations. All of these however, must be practical!

The control system should be well qualified to provide a continual flow of both process control and factory automation information. Along with a good flow of business information, such information makes possible the continual planning that is so essential to this dynamic paradigm.

The control system provides the visibility needed for the operator to properly understand the operations. As a conduit to the business plan, the link between that plan and the control system exposes the operator to appropriate portions of the business side of the change requirements. This will shift the operator's role from "running the stuff" to focusing on production optimization and product quality. The operator's job begins to become that of a business manager! (Figure 28-2)

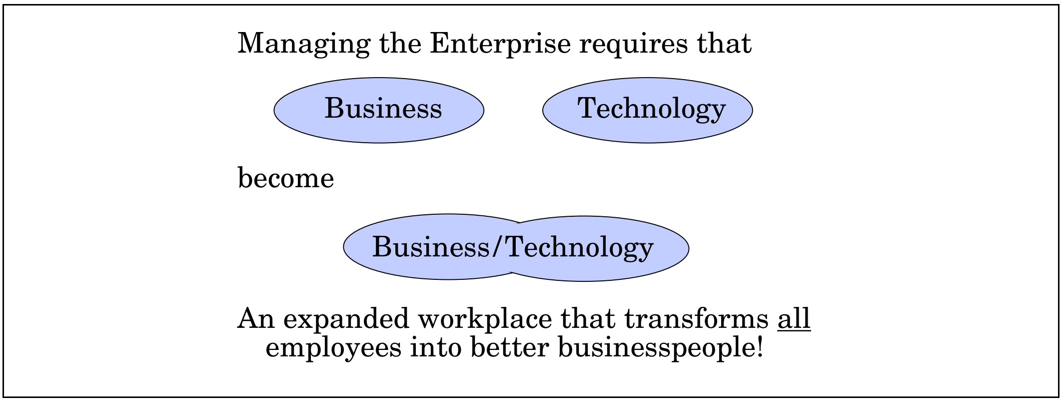


Figure 28-2. Technology is Changing Business!

Running a business, like operating a process, is an art!

There are no magic formulas that apply in every situation! Owners and operators alike must continually make judgments on when to apply known practices and when to be innovative. Both need tools to help decide what is the proper decision and when to apply that decision.

Computers cannot run a process or plant.

They are merely *tools* that help one perform the *art* of running a process or plant and do it *consistently.*

Good management of the whole enterprise requires an understanding of the business and the technology of that business. These two worlds have traditionally been separate for a variety of reasons. Today, however, successful corporations have been able to take advantage of the merger of both business and technology as the two become one. Because of this blending of business and technology, more information (not just data) is available to all parties involved, thereby expanding the workplace. This capability is truly empowering all employees, at all levels, transforming them into better business people.

Defining functions

A good typical example of any business is the batch operation (Figure 28-3). This is true in both the discrete and the process worlds. It is a matter of "scale." Simplistically, a continuous process is just a very long batch run. All of the elements of the batch operation are there. In fact, several "supporting" functions of the primary continuous operation are usually very much like traditional batches.

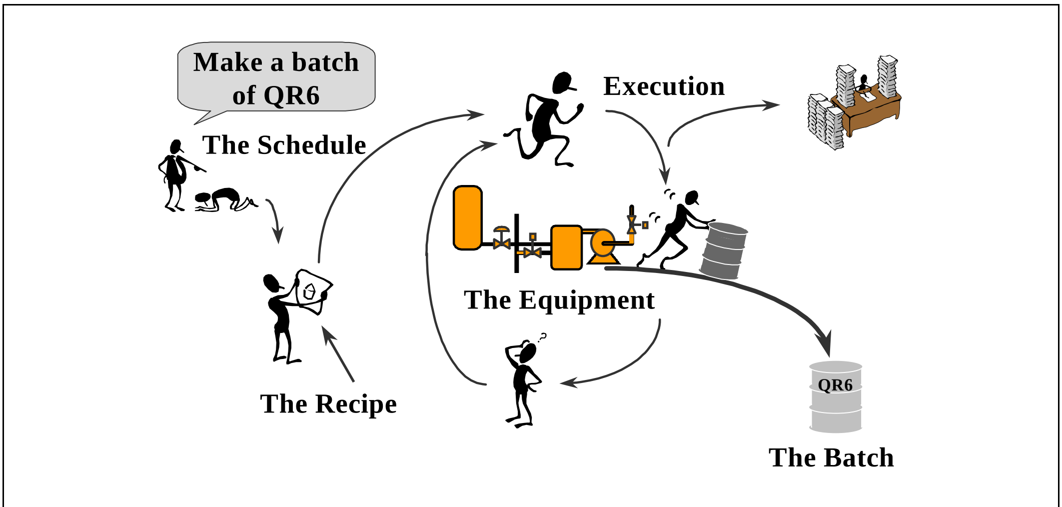


Figure 28-3. Components of a Typical Batch Operation Reflect Most Processes

Boiler constraint limiting control packages for large power utilities are every bit as complex as any chemical batch. This is also true of any metal heat-treating operation. Batch examples are found in steel mills, water treatment operations, paper mills, cement plants, glass making, and manufacturing. If you translate the terminologies, you discover the same problems. Of course, there are significant differences from one type of operation to another and from one type of plant to another. Usually, however, the differences are only in the degree of complexity.

Without going through the details of the emerging ISA standard S88 there are now formal definitions of every aspect of possible plant production (Figure 28-4) from the sensing and final elements up to (but not including) the enterprise (more precisely, the corporate business plan for the enterprise). There has been tremendous effort made to provide a modular product and process framework for the plant floor.

Many, however, see S88 as "just a batch control standard" instead of a general modular automation standard. There is no question that there is more that must be addressed quickly, now! At the ISA'96 Chicago Conference, the inaugural meeting of Committee S95 began to define enterprise/plant floor practices which must include explicit support all those issues in linking the business plan to batch, continuous, discrete, and hybrid plants.

Keep in mind, this link may extend remotely from corporate headquarters to all types of plants around the world (hence, "Enterprise"). It sounds so wonderful on paper, but it involves so many tortuous details!

The SP95 effort to cover enterprise/control functions has been motivated by the S88 "batch" standard, but it is most certainly very different:

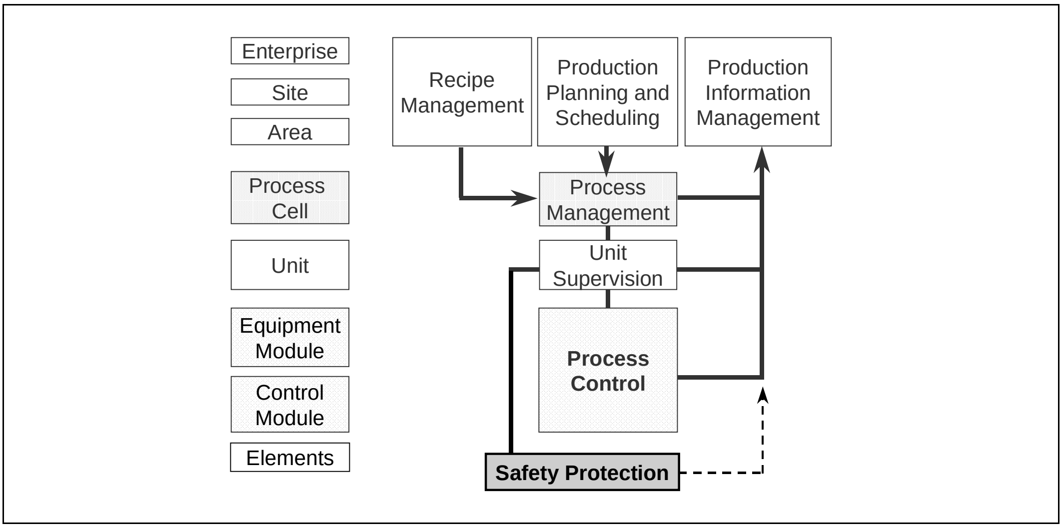


Figure 28-4. Batch Operations are Not Unlike all Continuous and Discrete Functions

S88 leverages *product* knowledge.

S95 leverages *business* knowledge.

Simplistically, one can envision a business plan being executed by providing explicit instructions to the plant floor and receiving back into the plan the results of each action. Of course, as the process becomes more complex and the number of connections increases, this approach becomes extremely impractical (Figure 28-5). Not the least of the problems is the "business computer" concept of "real-time" compared to the process concept of "real-time."

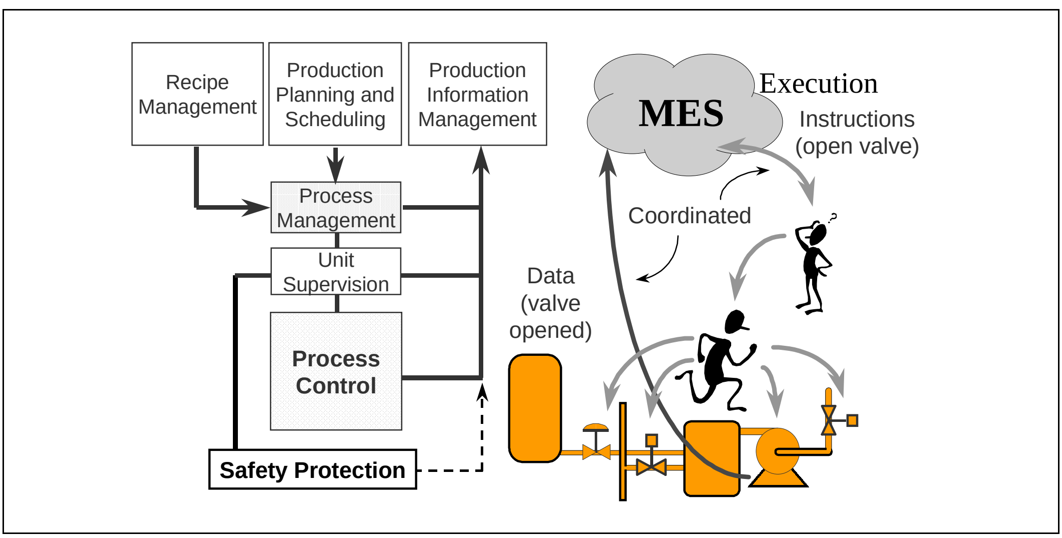


Figure 28-5. Business Plan Cannot Operate Each Device in the Process

It is just as simplistic to presume that an automation system improves conditions. Connecting the control system to the plant computer is nothing new, but currently *information* exchange is nearly always paper-based and then interpreted by individuals so as to meet the significance of the task at hand.

Where electronic connections are actually made, the solutions are very customized, and/or they are point-to-point links of specific data. Cause and effect are frequently mismatched, compared with the *real* needs of the enterprise (Figure 28-6). For example, the business plan doesn't really care which valves are opened or closed but rather is concerned with what is the current inventory of product and how is it changing. "Business computers" are not designed to rapidly shift priorities in response to plant upsets and to quickly analyze alarm spurts. Alarm spurts in a nine-story utility boiler control system at a power generation plant, for example, can come in at one thousand per second! Through appropriate control strategies that use today's technologies, savings occur within the automation system and *outside* the business plan, some may argue *instead* of the business plan. But it must be realized that control *strategies* are *different* than business plan *objectives*.

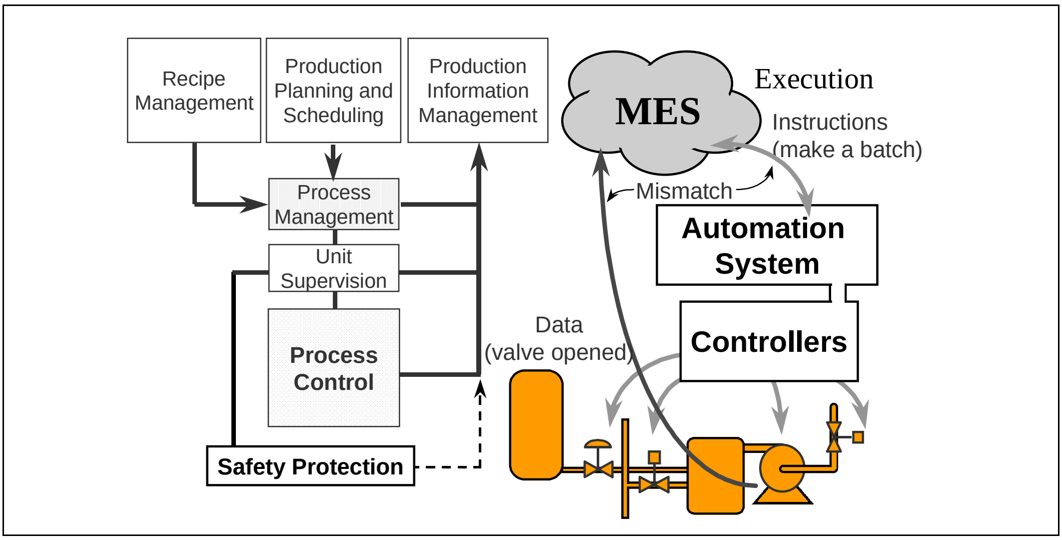


Figure 28-6. Adding Automation System Does Not Resolve Lack of Understanding

Business plan objectives: Save money by effectively allocating resources.

Automation system objectives: Save money by optimizing control strategies.

The two should not interfere with each other's goals, and creativity must be used so these goals can be achieved. There are two domains involved (Figure 28-7), that are fundamentally different:

- An informational domain that is transactional and involved in the search and analysis of information

- A control domain that is navigational and involved in causing action and observing the results

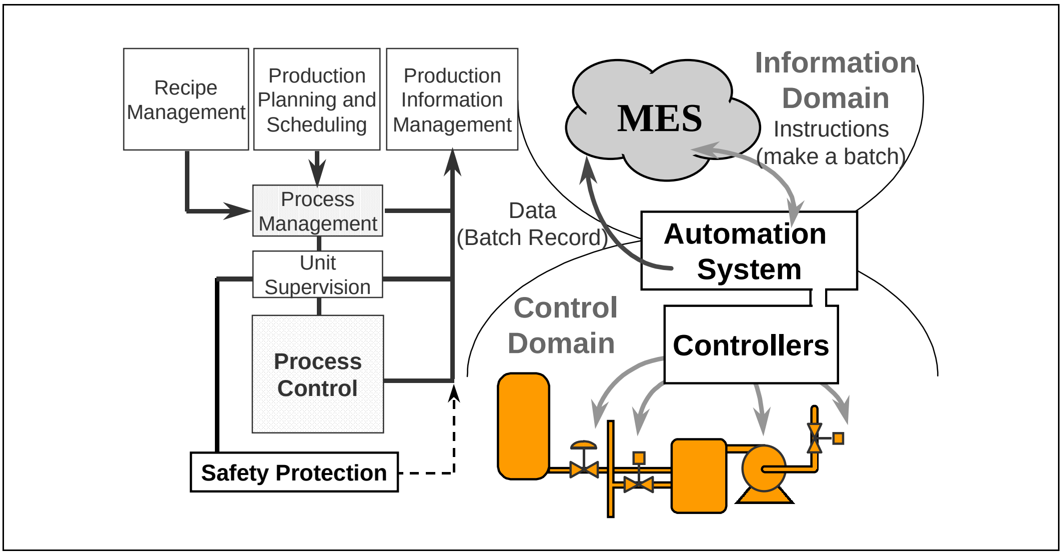


Figure 28-7. Boundary between Information and Control is *Not* Smooth!

During the past two decades, distributed control systems (DCSs) in the process world have evolved because of the increasing memory capacity and the increasing performance capability. This has resulted in a dichotomy of automation system. This system shares both control and no small amount of information (versus just data). Some rather sophisticated relational databases employ search techniques to allow us to analyze sort, compare, filter, qualify, and simplify raw data into information within the control system itself. Some of the results are already redirected to manage advanced control systems, with built-in constraints appropriate to the art of performing that particular process.

As a result, there is no easy demarcation between control and information domains but rather a "gray area." For example, the batch record is full of data, but it can be presented in such a way as to be information (i.e., data already predigested to reduce the search). This is the current battlefield of the many "middleware" vendors.

The Challenge of Two Computing Communities

All of this has set up some significant challenges to the SP95, not the least of which is bringing together two very different computing communities (Figure 28-8).

The businesspeople understand the complex interaction between such concerns as order entry, inventories, material procurement, scheduling, distribution, warehouse management, invoicing, and even (especially?) profit margin. They need copious record correlations that involve the annual, quarterly, monthly, daily, down to hourly tabulation of corporate-wide conditions of the equipment, personnel, and product. The

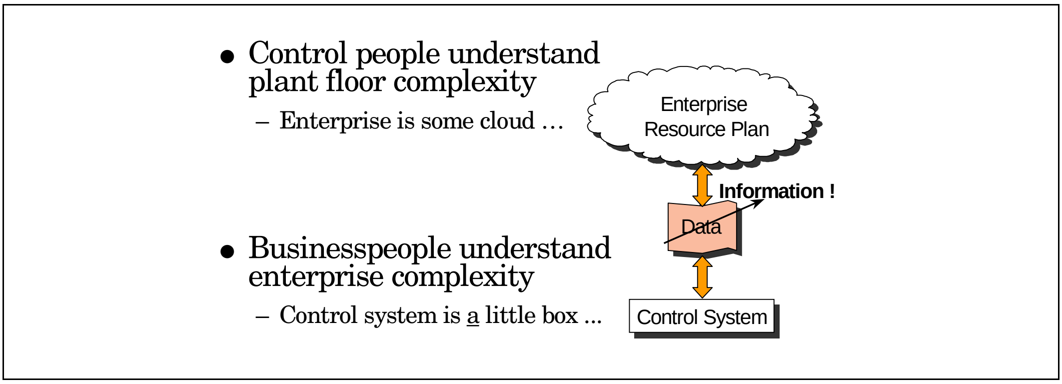


Figure 28-8. The Challenge Includes Combining Efforts of Two Different Computing Communities

control system is a single small box in their hierarchy of concerns. (But “no blockhead bit-nerd is gonna mess with my schedule and cost us profit margin.”)

The control people understand optimal production, deadlines, the strategy of creating product under continually changing conditions, dynamic regulatory agencies, environmental considerations, managing plant upsets, and analyzing alarm conditions while maintaining the safety of equipment, product, and personnel. They routinely need to respond to change within hours, minutes, seconds, milliseconds, and microseconds. The enterprise plan is some vague cloud that connects with their activities. (But “no stinking bean counter better mess with my set point and blow this plant to kingdom come.”)

Challenge of S95

The tasks of Committee SP95 are as follows:

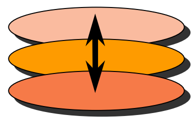
- Establish common vocabulary
- Find a common view of Problems
- Build a common model
- Define data structures

These tasks must be usable for both process and discrete manufacturing and unlike the years it has taken for so many other specifications this one must be completed promptly but thoroughly! S95 must allow the enterprise to become one with both the business process and the production process! Flexible production is a an increasing requirement of industry. Enterprise integration is extremely challenging. Dynamic requests that come from ERP must be accommodated by the control system. This is not just data mapping!

Just as S88 terminology, definitions, and concepts are slowly becoming known and useful outside the traditional batch industries, so S95 must also be applicable in all industries. The need for it is already overdue in continuous and batch processes as well as discrete manufacturing.

To properly operate any business, there needs to be good planning, appropriate tools for the execution of that plan, and correct controls for the implementation of that execution (Figure 28-9). A good control system must have proper feedback through out the operation to improve both the execution and the planning. As with any control system, the better this dynamic can be made to function, the better appropriate corrections can be made at all levels. It needs to be a *continuous* adjustment.

- **Business** Needs:
  - Planning [ERP]
  - Execution [MES]
  - Control [OCS]



- **Control System** Contribution:
  - Right information needed for business plan
  - Right timing when needed
  - Right context to be properly used
  - Right response to planning requirements

Figure 28-9. Successful Management of Business Requires Good Control of Production

The role of the control system is to provide the right information needed for the business plan at the exact time it is needed in the right context for it to be properly be used and with the right response to the requirements of the business plan! Without these essential ingredients, a viable business plan is not workable. Having the information (not just data) in the control system layer is not enough. There must be an easy and quick way to move this information to the other needed areas. The easier and more simple that link can be, the more likely it is that the correct information will be located and transported.

All of this requires a system that is readily altered to meet the changing needs in the plan and allows the plan to be easily changed to meet the marketing and distribution needs of the corporation.

Combining Business and Control Architectures

There are many corporate consultants offering methods for structuring businesses today. They frequently include several highly integrated business modules to meet the customized needs of many unique corporations in different industries. These business modules include tools for sales and distribution, purchasing, production, plant maintenance, financial accounting, human resources, customers, vendors, and so on. Are interconnected so that changes in any one function are correctly reflected in all the others to the appropriate scale.

One of these modules relates to process production; others include plant management, and so on. Connected to each of these is a powerful “point-and-click” link to the control system. This link is more than just data mapping but also includes diagnostics and memory (remember, the connection may be to other countries over phone, microwave, and the like).

Ideally, the workstation uses the same development tools as the link, so the link is *embedded* (Figure 28-10). This provides a completely integrated open control system (OCS) with *direct communication* to the planning software normal engineering, and with no added software beyond what customers need for their business applications. An additional benefit would be controllers that can act as a DCS, PLC, or both and a legally separated safety system that configures the same way as the controllers and can inherently communicate with the rest of the overall system.

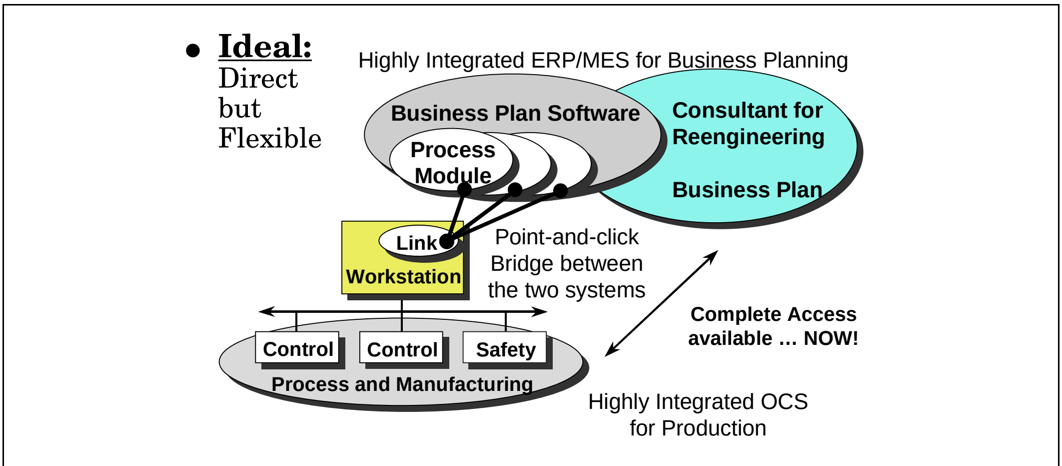


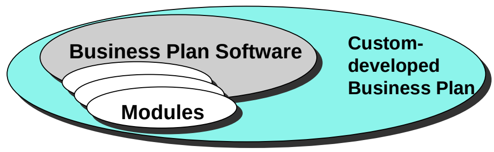
Figure 28-10. Creative Interconnection is Possible, but It Needs Disciplined Structure to Properly Manage the Enterprise

The business planning software vendor has invested its expertise in building the tools needed to efficiently run the business (Figure 28-11). These tools help to do the following:

- Manage *all* the resources the company has to do its job: machines and equipment, personnel, sales force, physical plant.
- Manage all the needed inventory: raw materials, storage, suppliers, finished-product handling.
- Manage the sequence and amounts of production: size of production runs, length of production runs, product mix, quality assurance.
- Manage all the scheduling activities: determining amounts and timing of deliveries from various suppliers, optimizing production with customer requests, providing "just-in-time" deliveries.
- Manage financial requirements: invoicing customers, paying suppliers, payrolls, performing cost analysis.

- Track business performance *with suggestions for change*: asset management, cost accounting, plant maintenance.

It is important to add here that the tools described here are best recommended by professional business consultants.



Role of Business Planning Vendor:
Managing the Business

- Resource management
- Inventory management
- Recipe management
- Scheduling

Figure 28-11. The Business Plan Needs Good Software Modules

The graphical mapping method presents side-by-side tree views of both the control system and the process module within the business plan (Figure 28-12). To map data, the user simply selects a process module field and visually attaches it to corresponding control tag variables. In a trigger window, the user defines the prompt conditions for the link to move data. These conditions can include any combination of time, events, ad hoc requests, or business plan requests.

This graphical mapping capability eliminates the time and expense of writing custom code by directly connecting the business plan to the plant floor. Because the software development tools are also the *foundation* for the workstation of the control system, no custom coding is needed to unite the control system with the link to the enterprise. The link is embedded.

Thus, the information flow between the control system and the process module of the business plan can begin in the early phases of an installation. In addition, if the business plan requires different information later, the control system can be easily reconfigured without requiring any hard-coded changes to the link itself, so the business plan can receive new data in minutes rather than months. In this way, the package permits faster start-up time, simplifies modifications, and removes roadblocks to responding to continuous changes of either the business plan or the control system.

For those interested in the technical implementation, note that:

In one approach to develop the link uses the business plan's remote function calls (RFC) to communicate to each of the business plan modules (Figure 28-13). RFCs can be used to communicate to any of the business plan modules, but the link certification currently covers only two. The shaded areas in Figure 28-14 represent the direct

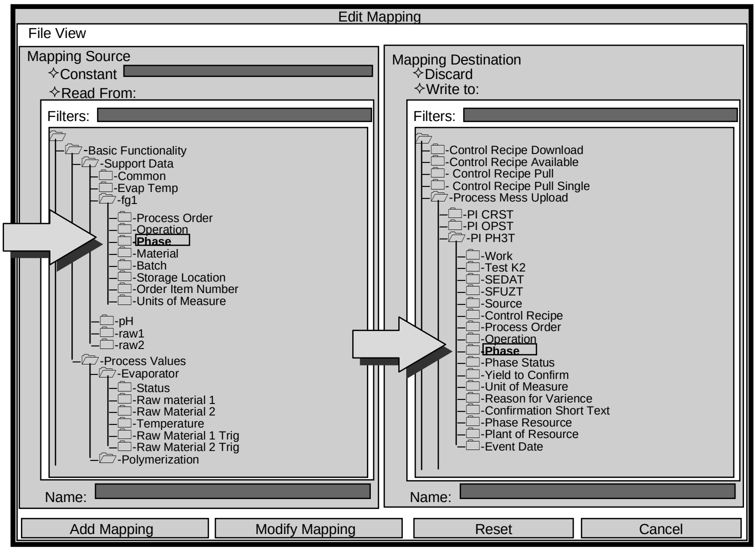


Figure 28-12. The Interconnection Requires Good Tools, Such as "Point-and-Click" Mapping, to Be Practical

connections made through the link to the open control system. The business plan software, which can be a very tightly integrated system, will already allow information to pass between other modules as needed by the enterprise.

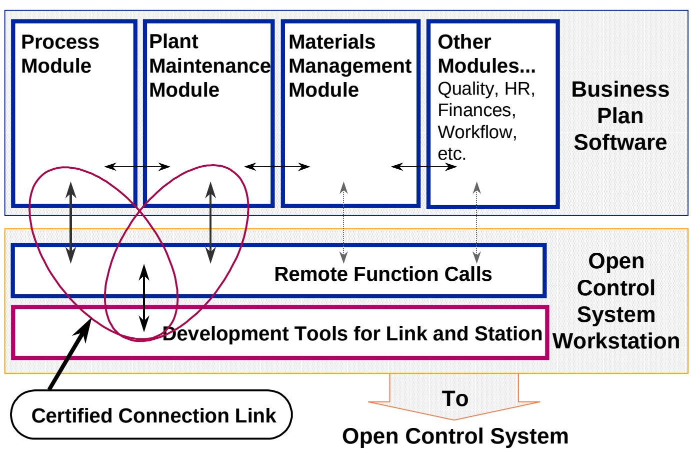
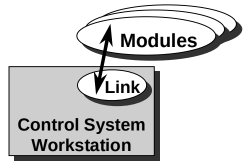


Figure 28-13. Software Components In One Approach to Implement the Coupling

The vendor of this middleware product integrates the business system with the control system (Figure 28-14). The object-oriented design reduces work and risk. It allows a visual mapping of the data between the process module within the business system and

Role of Link Application Platform Vendor: Connectivity



- Point-and-click mapping
- Mapping audit (import/export)
- Automatic event-driven messaging
- Simplification of module data structures
- Process data upload/recipe download

Figure 28-14. "Middleware" Products Ease the Pain of Integration

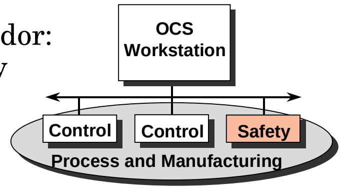
the control system, configuring a run-time server that uploads process data into the business plan and downloads recipes into the plant control system.

- This particular link provides the following:
- Point-and-click mapping between business modules and plant control systems.
- Flexible message triggering for process uploads and recipe downloads (Operation State, Phase State, Resource State, Material Consumption, etc.).
- A data server that automatically links business modules to plants.
- Data protection from network problems.
- Persistent data store for the information that the business module needs.
- Communication object that encapsulates business module and simplifies complex data structures.
- Communication object for software development tool supervisory software already within the control system vendor's product line.
- Interactive tracing and testing tools.

The control system must contribute to the productivity of the enterprise in many subtle ways (Figure 28-15). The control paradigm was established by control system suppliers at the interception of microprocessor-based control systems. Traditionally, two types of control systems are available: a DCS (distributed control system) and a PLC (programmable logic controller). The DCS was traditionally very good at controlling continuous processes that had very few discrete requirements, and the PLC was very good at logic-intensive control. Unfortunately, user applications are not split as cleanly as the suppliers' so-called solutions. Many applications have a mix of continuous requirements and discrete requirements if not on the application level then at least on the plantwide level. This presented the users with a dilemma when they were considering process automation. The following choices were available to the user:

1. A DCS in which *inexpensive* discrete functionality would have to be sacrificed.
2. A PLC in which *sophisticated* continuous functionality would have to be sacrificed.

Role of the
Control System Vendor:
Plant Control/Safety



- Saves time
  - Less configuration
  - No links to create
- Reduces costs
  - Less hardware to buy and install
  - Simple architecture to learn and maintain
  - Protects investment and avoids liabilities

Figure 28-15. The Control System Must Provide "Bottom-Line" Advantages

3. A Safety System, that is required to be separate from either option 1or 2.
4. A hybrid integration of these combinations which required implementing expensive interfaces.

Control systems emerged in the mid 1990s in which all of these functions were built in. Some can also configure all of these functions with any blend of the same languages as the IEC standard IEC #1131-3. This saves both time and expense.

OCS Requires More Productive Results

- Plant and Operator data is available from the beginning of project, for faster analysis into the business plan
- Can re-configure control to fit needs as they are discovered during start-up and operation
- Provides security of equipment, product, and personnel, specifically where needed without penaltyof installation and learning costs

More importantly, the control system provides more product results. It allows a faster start-up, enabling the control system to come online much faster. No additional cost and time for the engineering is needed to link the flow of information to the business system. Because of this faster availability, information can flow into the business system earlier. Analysis will always suggest some changes to be made in the control strategies (that is why the system was purchased!). As these needs are discovered, they can easily be incorporated into the control system. Ongoing business analysis will continue to study the plant's performance, and suggestions for modifications will continue. The flexibility of the control system enables it to always embrace new improvements as they present themselves.

As we mentioned earlier, for a business to be viable, it needs (Figure 28-16) planning, execution and control:

Planning is done using the business planning software packages with the highly integrated modules that are appropriate to your plant(s), and by communicating through the process module of that software.

Execution occurs through the simple, highly integrated "point-and-click" interface within the control system workstation.

Control is achieved through the highly integrated open control system and managed through the same control system workstation.

Total System Must Meet Business Needs:

- **Planning** Business planning software
- **Execution** "Point-and-click" linking software
- **Control** Open control system

- Must be easy to use.
- Must have consistent openness.
- Must have workable domains for both business and control functions.

- **Must work as ONE system!**

Figure 28-16. Requirements for Good Enterprise Resource Planning (ERP) System

This permits a system that is unified with the business plan and capable of being responsive to it but also capable of implementing the emerging requirements of S95. All of these ideas are useless, however, unless all the effort and capabilities are blended to bring about a system that performs as one!

Plant Information 29

One of the business dilemmas is optimizing the making of a product and determining the costs of that optimization. On the plant floor, the concern is how you manage the operation and stay within temperature, pressure, and flow limits and how you relate those limits to the quality of the product. This is an entirely whole different type of report and a whole different kind of a concern for control room operators on the one hand and the plant managers on the other. Now, it's a fact that the business's information needs require data from the plant floor for some of the calculations, such as energy consumption and inventories of raw materials or finished product. On the plant floor the concern is information that's dealt with in a matter of minutes, or seconds or even microseconds, whereas on the business side you're dealing with information that is gathered over weeks and months and quarters or even a year or so.

Flow of Data

Corporate objectives must be transmitted through the plant level operations down to the plant floor and ultimately to the controllers and final elements (Figure 29-1). Sensors in the unit processes transmit data back up through the structure and the results are analyzed to determine the results of that planning. Each of these layers of computing functions needs access to databases in each of the other realms. These functions do not reside in neat little compartments, nor are they located within their own computers. In the real world, most plants "just grew" with now outdated concepts, plans, and computers. They were correct at the time, it's just that the changing technology has allowed so many more considerations to come into play. Nevertheless, databases of data need to be addressed.

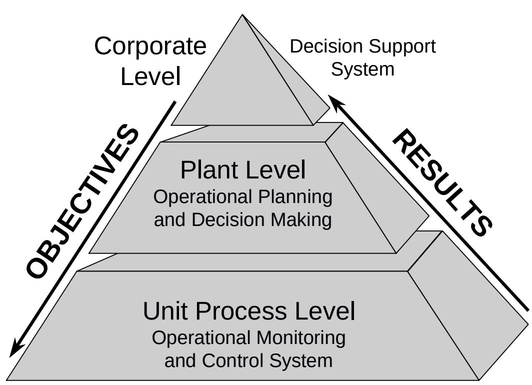


Figure 29-1. Computing Layers of Process Information Flow

Here we are with our hierarchies again, but now with new terminologies (Figure 29-2). These realms, shown in Figure 29-2, have been defined by the Gartner Group, a market research firm. They are generally accepted by most people in the late 1990s, but exactly which functions reside in each is open to a variety of opinions. For other definitions in this field, see boxed sidebar at the beginning of Chapter 28.

- ERP
  - Enterprise resource planning systems
- MES
  - Manufacturing execution systems
- OCS
  - Open control systems

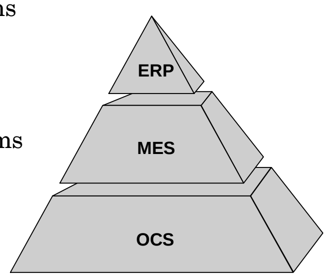


Figure 29-2. Gartner Group Definitions of System Layers

- Enterprise Resource Planning Systems (ERP)—Top-level software that contains Forecasting, Planning, Sales Order, Receiving.

- Manufacturing Execution Systems (MES)—This is the Middleware that contains Scheduling, Reporting, Document Management.

- Open Control Systems (OCS)—OCS is a general term for any open control system including DCSs, PLCs, or PCs (Personal Computers). The implication is their complete connectivity of controls with all other networked devices, which for most control systems in the 1990s is quite optimistic.

MES is a buzzword created a few years ago to describe the nebulous layer of software that connects your business systems (ERP) to your control systems (OCS). There are may independent packages that do this. Nevertheless:

1. Many ERP vendors provide MES functionality.

2. OCS vendors also provide some MES functionality (Dispatching, Reporting).

There are some that feel this MES “layer” will disappear, but in fact the FUNCTION will remain, wherever it is physically located.

OCS is the realm of automatic control functions, which include supervision, continuous and discrete control, monitoring, and alarming. The real challenge in this realm is working with real-time data manipulation.

Timing is Everything

> Computers were never real-time at best, some are "real-enough time."

As we see in Figure 29-3, memory capacity, processing speeds, functions needed, all vary with the "level" in the hierarchy of the plant. In the past, this has required many different computers, some connected with others, many requiring that the data be reentered by some clerk after it was analyzed for the function involved. We also see, however, that

- The typical business database manager system has high levels of human transactional access requirements by users; for example, a relatively large slow-changing volume of money in a bank but continually accessed by many hundreds of tellers (operators).
- The typical control database manager system has very low levels of human access requirements by users; for example, one or a few operators accessing a large database of hundreds of rapidly changing sensors and final elements in a process.

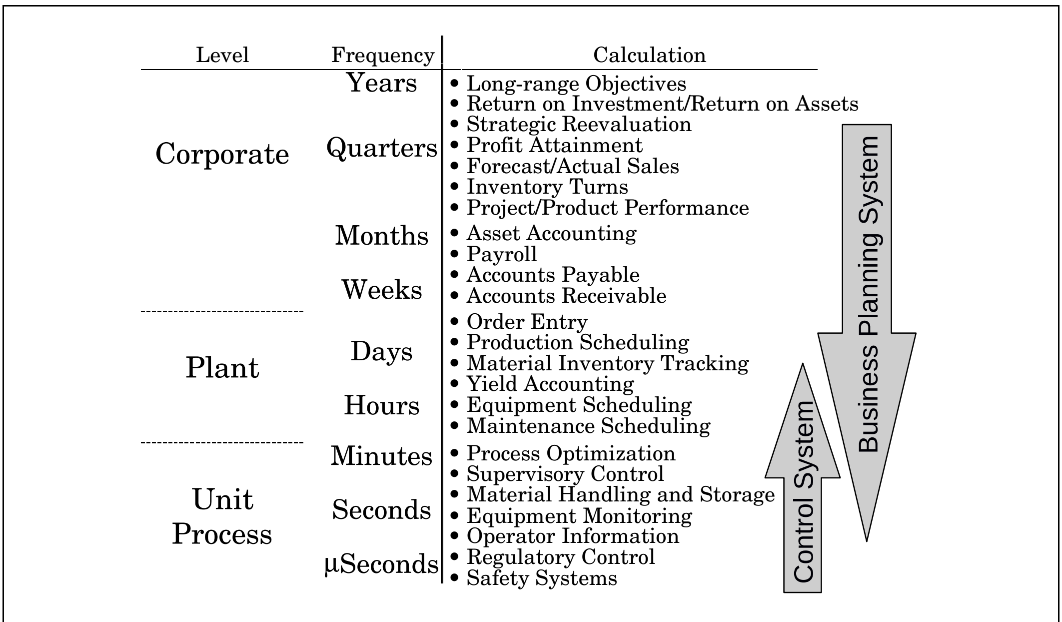


Figure 29-3. There is an Overlap of System Functions; No Clear Line of Separation

So we then see in Figure 29-4 that the database manager systems (DBMS) for business and for process control have totally different orientations. From a software standpoint, they would not seem merely upside down, but even inside out.

These two distinct types of databases—one of speed and rapid manipulation on the plant floor and the other of capacity of data with a much slower need to know—require different kinds of database structures and different kinds of data manipulation. The

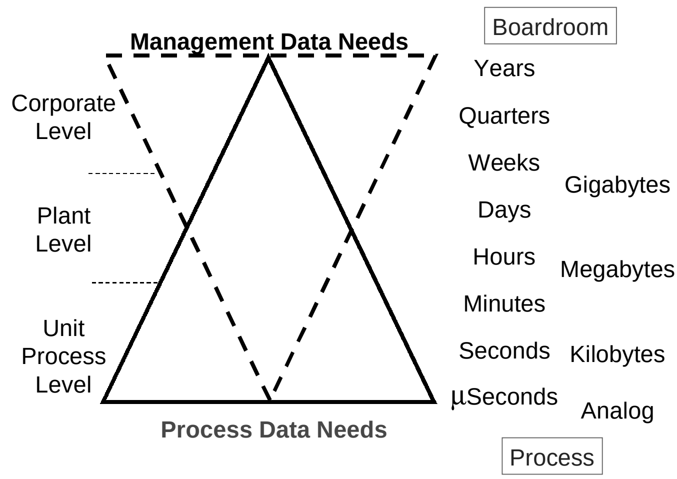


Figure 29-4. DBMS for Process and for Business Have Very Different Orientations

differences are between working with transactional data on the business side and with analytical data on the plant floor. Now, of course, there is analytical data on the business plant side as well as the upper side of the structure, but it is of a different form and timing.

The easiest way to contrast these, is by viewing an inverted pyramid or delta, for the transactional data compared with a right-side-up delta for dynamic process analysis (Figure 29-4). The base of a delta has hundreds, maybe thousands, of inputs and outputs to the control system, and the point at the top is typically a single or small group of operators managing and manipulating that data. In a business structure, it is an inverted delta, and the broad base is really at the top where you may have many, many people such as the tellers of a banking system and the point is at the bottom where there is a core of money. This pot of money changes relatively slowly in size but a lot of daily transactions are continually registering with all of the teller machines, tellers, and people with screen views at the top end.

Data Management

Database management software is one of the largest software categories. Today, almost every user of a computer, whether it is a PC, a minicomputer or a main frame, make use of this kind of software to some degree. From keeping lists of events and alarms, entering batch recipes, and analyzing trends of operations to the modeling of an entire business strategy or enterprise, the data required for these operations is typically stored in and accessed from a database management system (DBMS).

Two main areas of business are online transaction processing and online analytical processing. The latter is colloquially referred to as “end user computing.” For most online processing applications today, a DBMS based on a relational model is often used. Nevertheless, online analytical processing has a distinct requirement different from the traditional relational database manager. As a result, a multidimensional model is emerging.

Many users of the relational type of DBMS find that it handles some problems rather poorly. During dynamic events in process operations, for example, users need to analyze the situation. This involves more than just recording and retrieving large amounts of information about the process and production. It also differs from order entries or marketing, sales, and financial data. Because of this, other types of DBMS's are gaining popularity, especially multidimensional DBMS products. In an era where computers are proliferating and end users want to use them as effectively as possible, a single DBMS approach is impractical. It cannot satisfy all the computing needs of an entire business.

Using the wrong kind of DBMS raises the cost of performing a task and slows the retrieval of information (rather than data). The DBMS may also require several layers of add-on software before it can work the way users want it too. In some cases, it may not even be possible for the DBMS to perform the task desired. The reason it is hard for a single data base system to perform all the tasks now required of it stems from the differences between transaction and analytical process applications.

In Figure 29-5 we see the parent-child relationship of a hierarchical database, which requires that the system trace its way from one leg to another, which consumes time and processing power. Queries must be made in the same way data was stored, more suitable for fixed transactions. Transactional applications are concerned with current, ongoing data. The record of an order is often deleted soon after product is shipped in some industries, for example. Exceptions include pharmaceuticals and heat treating. Marketing departments may keep portions of the records as historical data for future projected. Data may be kept back for two or more years so it may be inspected for trends and forecasts. Transaction applications are generally static, whereas plant control applications are dynamic. For instance, once an accounting system has been designed and implemented, the files or fields are rarely changed, and the transactions and screens used to enter them rarely change either. In control applications, however, it's harder to plan the data structures ahead of time and predict all the ways in which the data will be used.

Users in process control operations need maximum flexibility, good presentation of the data from every imaginable perspective, and the option to create new screens on the fly. Transaction applications are usually flat or file oriented and consists of linear lists of related values, each describing an employee, customer, dates, product and similar entities. Most of the time, this data is accessed by single key value and is viewed as simple list on the screen or a page. The flat relational database management system (RDMS) of Figure 29-6 acts more like a lookup table, allowing queries from the database to be different from the way the data was stored. Dynamic changes are difficult, and all parameters must always be available for every situation. Process control situations, in contrast, are best viewed in a multidimensional format because the user needs to look at all the possible combinations and interrelationships.

Although most end-user applications are done at the overview level, they also require the ability to drill down into the detailed data from which that overview was derived. This "drill down" may follow a hierarchical order that unfolds more and more detail as needed. All told, the differences between transaction and end-user applications are so many and so large that the effects of a specific DBMS software approach on performance and user satisfaction can be quite dramatic. To be effective, the DBMS software should fit the job. To do this, the design should focus on three areas: data model, data access and manipulation language, and implementation strategy.

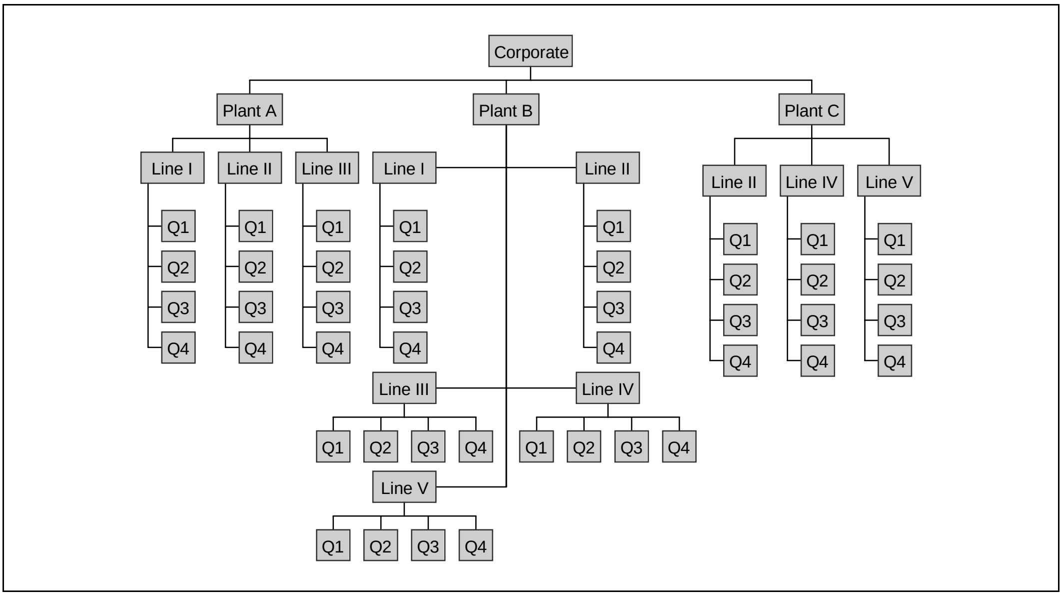


Figure 29-5. The Parent-Child Relationship of the Hierarchical Database

Plant	Line	Quarter	Production......
A	I	1	
A	I	2	
A	I	3	
A	I	4	
A	II	1	
A	II	2	
A	II	3	
A	II	4	
A	III	1	
A	III	2	
A	III	3	
A	III	4	
B	I	1	
B	I	2	
B	I	3	
B	I	4	
B	II	1	

Figure 29-6. Flat RDMS Acts More Like a Lookup Table

The DBMS data model is the logical representation of how the data is structured and how operations will be performed on it. It is the principal determinate of how the user will have to look at the data. The so-called single file data model of record sharing the same structure is still widely used. Data models of greater sophistication are also widely used for transactional applications. The hierarchical model is based on a parent-child relationship between different record types. Some of the current popular relational models are really an extension of the single file data model. It gives a database of many single files the ability to dynamically combine the data across different files. This ability is important because of the types of data that need to be combined frequently.

The data model of Figure 29-7, in which data is viewed in multidimensional arrays, shares some features with the relational data model but is oriented much more toward the data interrelationships. In the relational model, these are rarely explicit and so must be defined in some way by the end user. It is these interrelationships that give the multidimensional data model its power to support end-user applications, most of which are concerned precisely with understanding those relationships. In this data model the basic unit is dimension. Essentially, a dimension is a set of similar entities, such as the set of all of the alarms associated with a single unit or all of the temperature profiles of a particular device. The data is structured by these dimensions into arrays with a vertical dimension, a horizontal dimension, a page dimension, and so on. The contents of these arrays are actual values.

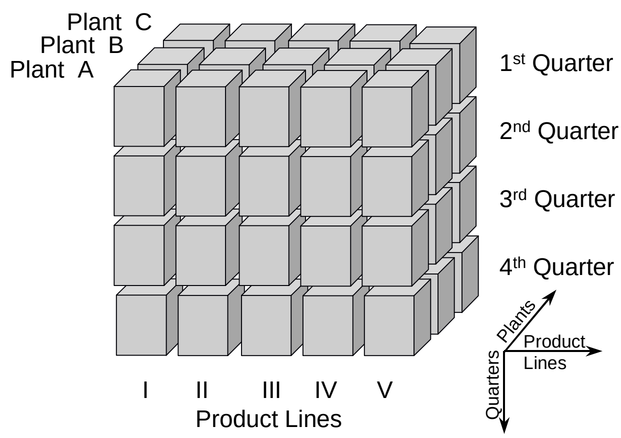


Figure 29-7. Flexible Multidimensional Database Provides Better Views of Relationships

This more flexible multidimensional database uses less memory, has faster access, and provides better views of relationships that were not anticipated when gathered.

Operators typically want to know what the data represents then its record structure. That is, an operator wants to know what is happening in a process or, more importantly, what is happening in the yield of the product, but rarely wants to know all the temperatures over the last shift. More likely, he or she wants to know the interrelationships between temperatures, pressures and flows, particularly at times when different operations occur rather than just a straight-line calculation of everything that happens in that shift.

When analyzing a process yield, for example, the engineer will want to know all of the common things that happened on the different times that this particular product was put together. He or she may want to trace the half-hour prior to the end of the product in each of five different occasions, each of which may have been spread out over the past three months. This will give the engineer a pattern from which to determine what the parameters are influencing the quality of the product. A pure list of temperatures would not give him or her this answer.

An interesting component of the multidimensional model is the singularity of dimension; that is, the product dimension exists once, and only once, for the whole database, no matter how many arrays there are that the product participates in. Each entity, or item, in a different dimension is identified by a unique alphanumeric key. The key lets the user automatically access all the information about an item through the entire database not just within a single file, as it would in a relational database.

Multidimensional views reduce storage requirements. Since data in a relational database is combined only through the matching of common keys across many files, keys can proliferate, greatly increasing storage needs. Multidimensional databases reduce the control data redundancy because keys are stored in single central lists, the dimension itself. For example, let's take a plant in which there are many, many pieces of equipment used in various combinations to make a wide range of different products (Figure 29-8). This could be a batch operation where it is often just as important to know which portions of the equipment have no data and which do. A user could determine that piece of equipment in which a particular type of product is *not* being made as well as that in which it is.

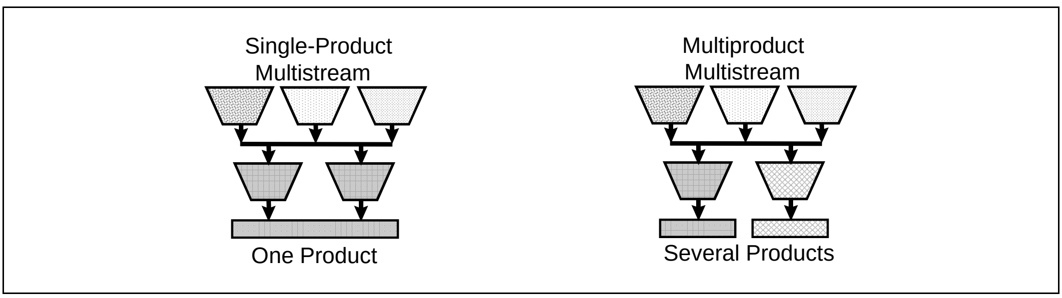


Figure 29-8. Database Parameters Are Continually Changing in Some Processes

In a batch operation, there are often many kinds of product passing through many different vessels. This can vary based on which vessels are being cleaned, which are being used for other production cycles, and so on. All the different product do not need the same parameters measured. If a vessel is empty, there are no temperatures, pressures, and flows, so there is no need to store all of the information that is not there. Now, many implementations of a relational system cannot readily change what is being saved in an active way while production is running. In these systems, it is difficult to refine or add attributes or parameters that become apparent after you create the product.

The multidimensional approach, on the other hand, allows dimensions to be maintained independently, making it easier to visualize what is missing. In other words, some products may need temperatures and pressures to be monitored, while others may need pressures and pH monitored. Still others may need flows and temperature combinations monitored. To retain data on all those parameters of temperatures, pressures, and flows, and pH would be a waste of memory. What is needed is to be able to save the parameters necessary for each product, and selectively save them and to make a new product that needs to measure different parameters, like consistency or weight. This requires a system that can add these parameters without losing or changing the data that went before it. We can combine the impact of alarm handling across all of these situations. Particularly useful is using this approach in circumstances where alarms are not likely to occur but later using a combination of equipment and product where alarms do occur. Furthermore, alarms do occur, you need to know to what extent and in what parameters.

Manipulating Data

The data manipulation language (DML) is the means by which the database manager system (DBMS) is instructed to store, retrieve, subset, match, calculate, or otherwise manipulate the data. You may wish to use a manipulation language to ask a DBMS to find all the products whose total temperatures have exceeded a certain value. Not surprisingly, the DML employed by a multidimensional database differs markedly from the standard query language (SQL) of a relational database. Record-oriented languages, such as an SQL involves specifying the attributes to be retrieved and the tables or files to be searched. Records are identified by means of search predicates for example all the temperatures over 100° C. That will filter the data so you can determine which individual records are returned by your query. In operation, a record-oriented query often causes the system to scan many or even all the records in the participating database tables.

A multidimensional DML, on the other hand, is a domain-oriented language. The domain is synonymous with dimension in this context. In this case, the emphasis is on selecting the parameters of interest, such as the products, their various temperatures, or the time periods involved, without regard to the files in which they appear. The idea is that the DMBS scans the various components in the indicated domains or dimensions, pulling together on the fly whatever information is needed. A case in point might be the yield point of a product over a period of history as compared with the quality yield over the next quarter of operations.

The differences in the need for business management-type software as opposed to data collection and reporting are a matter of trying to determine the many relationships between the following:

1. The amount of inventory that must be stored to create a product,

2. The amount of inventory that must be stored after creating the product before delivery,

3. The amount of energy consumed in the process of creating the product, and

4. The ideal implementation of various control strategies for determining the best methods for developing a high-quality product.

More Reality

We said earlier that in real-time processing, unlike typical business processes, in real-time processing, you have some requirements for speed and capacity that are far different than what occurs in the business world. Instead of large data tables, you have a lot of instantaneous actions from the various sensors that must be processed. And, at the same time, you have to manage the calculation of the totals, inventories, and quality that must go into the larger computer system "above" this activity to manage the business and develop a business plan. Because of all these different requirements, you require different strategies, which gets us back to the multidimensional databases versus the relational databases.

In either case, there is no free lunch. The strategies for updating a multidimensional database are much more expensive than for updating a typical relational system. This is because a large number of arrays must be searched for individual entries in order to update a transaction record. It is generally not realistic to use a multidimensional database manager system (DMBS) to perform a function for which it was never intended, namely on-line *transactional* processing on the plant floor. Likewise, relational databases though successful at many transactions and simple query applications on the plant floor have not proved themselves in many end-user computer applications on the business plan side. So when deciding which database manager technology to employ, the potential user should not ask simply which database is best, but rather, which is best suited for the type of applications that must be created.

Index