

---

# SECTION 11

---

# STANDARDS OVERVIEW\*

---

## **Terry Blevins**

*Fisher-Rosemount Systems, Inc. Austin, Texas (Fieldbus)*

## **Thomas G. Fisher, P. E.**

*Lubrizol Corporation, Wickliffe, Ohio (Batch Control)*

## **Paul Gruhn, P. E.**

*Moore Products, Houston, Texas (Safety Instrumented [Interlock] Systems)*

<b>SAFETY INSTRUMENTED (INTERLOCK) SYSTEMS</b>	<b>11.3</b>
<b>INTRODUCTION</b>	<b>11.3</b>
What People Really Want	11.3
What the Standards Actually Are	11.3
<b>DESIGN LIFE CYCLE</b>	<b>11.3</b>
Conceptual Process Design	11.4
Hazard Analysis and Risk Assessment	11.4
Application of Non-SIS Layers	11.5
Is an SIS Required?	11.5
Define THE Target Safety Integrity Level	11.5
Develop Safety Requirements Specification	11.5
Conceptual SIS Design	11.6
Detailed SIS Design	11.6
Installation and Commissioning	11.7
Operations and Maintenance	11.7
Modifications	11.7
Decommissioning	11.7
<b>MULTIPLE INDEPENDENT SAFETY LAYERS</b>	<b>11.7</b>
Process Plant Design	11.7
Process Control System	11.8
Alarms and Operators	11.8
Shutdown/Interlock Systems	11.9
Fire and Gas Systems	11.9
Containment Systems	11.9
Evacuation Procedures	11.9
<b>SYSTEM TECHNOLOGIES</b>	<b>11.9</b>
Logic Systems	11.9
Field Devices	11.10
Sensors	11.10
Final Elements	11.11
<b>SYSTEM ANALYSIS</b>	<b>11.11</b>
Notes for Table 1	11.12
<b>KEY POINTS</b>	<b>11.13</b>

---

\* Persons who authored complete articles or subsections of articles, or who otherwise cooperated in an outstanding manner in furnishing information and helpful counsel to the editorial staff.

<b>RULES OF THUMB</b>	<b>11.13</b>
<b>REFERENCES</b>	<b>11.13</b>
<b>AN OVERVIEW OF THE ISA/IEC FIELDBUS</b>	<b>11.14</b>
<b>INTRODUCTION</b>	<b>11.14</b>
<b>PHYSICAL INSTALLATION OF A FIELDBUS SYSTEM</b>	<b>11.15</b>
<b>UTILIZING FIELDBUS DEVICES TO MEET</b>	
<b>APPLICATION REQUIREMENTS</b>	<b>11.21</b>
<b>DIAGNOSTIC SUPPORT OF FOUNDATION FIELDBUS DEVICES</b>	<b>11.23</b>
<b>CONTROL SYSTEM IMPACT</b>	<b>11.27</b>
<b>EXAMPLE INSTALLATIONS: COMMERCIAL</b>	
<b>FIELDBUS INSTALLATIONS</b>	<b>11.29</b>
<b>ESTIMATING SAVINGS FROM USING FIELDBUS TECHNOLOGY</b>	<b>11.30</b>
Reduction of Terminations and Home Run Wiring	<b>11.31</b>
Reduction in the Number of I/O Cards	<b>11.32</b>
Reduction in Instrument Room Space	<b>11.33</b>
<b>SUMMARY</b>	<b>11.33</b>
Best Practices in Applying Fieldbus	<b>11.34</b>
<b>REFERENCES</b>	<b>11.35</b>
<b>BATCH CONTROL: APPLYING THE S88.01 STANDARD</b>	<b>11.35</b>
<b>INTRODUCTION</b>	<b>11.35</b>
<b>DEFINITIONS</b>	<b>11.36</b>
<b>RECIPES</b>	<b>11.37</b>
Recipe Types	<b>11.37</b>
Recipe Information Categories	<b>11.38</b>
<b>EQUIPMENT ENTITIES</b>	<b>11.40</b>
Equipment Control	<b>11.40</b>
Physical Model	<b>11.40</b>
Partitioning Equipment Entities	<b>11.42</b>
Procedural Control Model/Physical Model/Process Model	
Relationship	<b>11.43</b>
<b>RECIPE PROCEDURE/EQUIPMENT CONTROL SEPARATION</b>	<b>11.44</b>
Control Recipe Procedure/Equipment Control Linking	<b>11.46</b>
Control Recipe/Equipment Procedural Elements	<b>11.47</b>
<b>PROCESS AND CONTROL ENGINEERING</b>	<b>11.47</b>
<b>CONTROL SYSTEM FUNCTIONAL SPECIFICATIONS</b>	<b>11.49</b>
What Is Needed to Define Batch Control	<b>11.49</b>
Equipment Entity Details	<b>11.51</b>
<b>SUMMARY</b>	<b>11.56</b>
Key Points	<b>11.57</b>
Rules of Thumb [6]	<b>11.58</b>
<b>REFERENCES</b>	<b>11.58</b>

# SAFETY INSTRUMENTED (INTERLOCK) SYSTEMS

by Paul Gruhn\*

## INTRODUCTION

---

This article provides an overview of the ISA Standard on Safety Systems (S84), with an emphasis on the layers of protection and the estimates of the availability of different types of instrumentation and controls. The definition of a safety instrumented system is a system designed to respond to conditions of a plant, which may be hazardous in themselves, or if no action were taken could eventually give rise to a hazard. It must generate the correct outputs to prevent the hazard or mitigate the consequences.

The ISA (International Society for Measurement and Control) S84 and IEC (International Electrotechnical Commission) 61508/61511 standards, along with the AIChE CCPS (American Institute of Chemical Engineers, Center for Chemical Process Safety) Guidelines on safety instrumented (interlock) systems, as well as process safety management legislation [1]–[4] are performance oriented, not prescriptive. They do not tell people what technology logic system to use (relay, solid state, or software based), what logic and field device configuration to use (single, dual, or triplicated), or how often to test a system (monthly, quarterly, or yearly). They merely list the performance requirements for the overall system. In other words, the greater the level of risk of the process, the greater the performance needed of the safety instrumented system.

### What People Really Want

However, what most people really want is a simple “cookbook” of preplanned solutions. For example, for a refinery, turn to page 35 in standard ABC. There it shows dual sensor, dual logic, simplex valves, yearly test interval, and so on. For an offshore platform, turn to page 63. There it shows. . . . Unfortunately, at this point in time, industry standards are not written this way. The standards do *not* give clear, simple, precise answers. They do *not* mandate technology, level or redundancy, or test intervals.

### What the Standards Actually Are

There is a fundamental change in the way industry standards are being written. Standards are moving away from prescriptive standards and toward more performance-oriented requirements. After all, it’s relatively easy to be prescriptive about something we have a great deal of experience with (e.g., boilers). The same cannot be said of relatively new and unproven processes. This means each plant will have to decide for itself just what is safe, and each plant will have to decide on how it will determine and document that its systems are, in fact, safe. Unfortunately, these are difficult decisions that few want to make, and fewer still want to put in writing.

## DESIGN LIFE CYCLE

---

Designing a single component may be viewed as a relatively simple matter, one that a single person can handle. Designing any large system, whether it’s a car, a computer, an airplane, or a safety instrumented system, however, is typically beyond the ability of any single individual. Large systems

---

\* P.E., Moore Products, Houston, Texas.

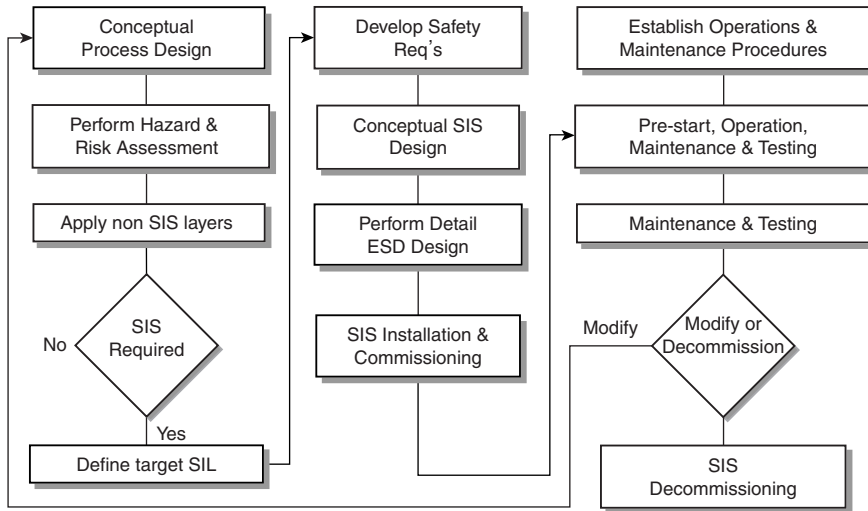


FIGURE 1 ISA S84 design life cycle.

require a multidiscipline team. The control system engineer should not feel that the entire burden of designing a safe plant rests on his or her shoulders alone, because it does not.

Experience has shown that a detailed, systematic, methodical, well-documented design *process* is called for in the design of safety instrumented systems. This starts with a safety review of the process, implementation of other safety layers, systematic analysis, and detailed documentation and procedures. The steps are described in most documents as a safety design life cycle. The intent is to leave a documented, auditable trail, and make sure that nothing is neglected because it has fallen between the inevitable cracks within every organization.

Large systems require a methodical design process. Figure 1 shows the life-cycle steps as described in the ISA S84 standard. This should be considered one example only, as there are variations of the life cycle presented in other industry documents. A company may wish to develop its own variation of the life cycle based upon its unique requirements.

Some will complain that performing all of the life-cycle steps, like all other tasks designed to lower risk, will increase overall costs and result in lower profitability and productivity. One in-depth study in the past, conducted by a group including major engineering societies, 20 industries, and 60 product groups with a combined exposure of over 50 billion hours, concluded that *production increased as safety increased* [5].

### Conceptual Process Design

The first step in the life cycle is to develop an understanding of the process, the equipment under control, and the environment (physical, social, political and legal) in sufficient depth to enable the other life-cycle activities to be performed. The goal is to design an inherently safe plant. The activities in this step are generally considered outside the realm of the control system engineer.

### Hazard Analysis and Risk Assessment

The next step is to develop an understanding of the risks associated with the process. Risks may impact personnel, production, capital equipment, the environment, company image, and so on. Hazard

analysis consists of identifying the hazards. There are numerous techniques one can use (HAZOP, What If, Fault Tree, Checklist, etc.) and numerous texts describing each method [6]–[8]. Risk assessment consists of classifying the risk of the hazards that have been identified in the hazard analysis. Risk is a function of the frequency or probability of an event, and the severity or consequences of the event. Risk assessment can be either qualitative or quantitative. Qualitative assessments subjectively rank the risks from low to high; quantitative assessments, as the name obviously implies, attempt to assign numerical factors to the risk, such as death or accident rates. This is not intended to be the sole responsibility of the control system engineer. There are obviously a number of other disciplines required in order to perform these assessments.

## Application of Non-SIS Layers

The goal of process plant design is have a plant that is inherently safe, or one where residual risks can be controlled by the application of noninstrumented safety layers. KISS (keep it simple, stupid) should be an overriding theme.

## Is an SIS Required?

If the risks can be controlled to an acceptable level without the application of an instrumented system, then the design process stops (as far as a safety instrumented system is concerned). If the risks cannot be controlled to an acceptable level by the application of noninstrumented layers, then an instrumented system will be required.

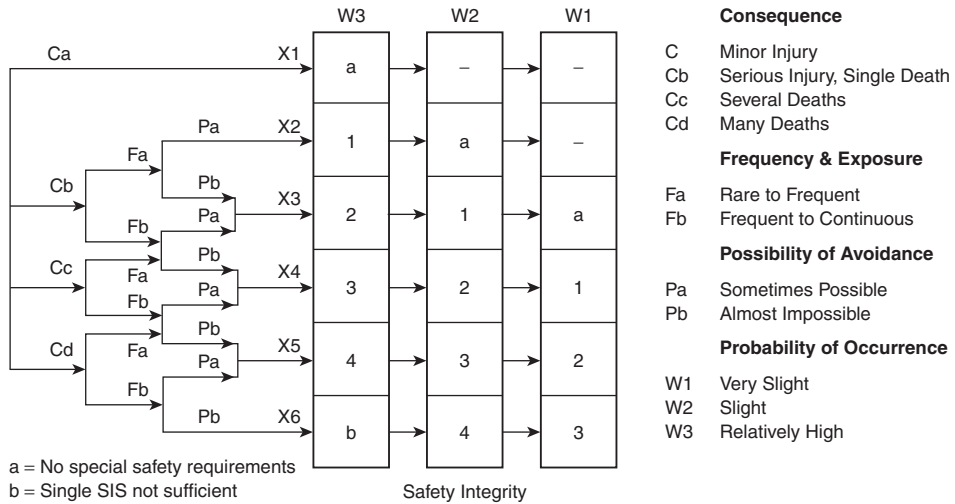
## Define the Target Safety Integrity Level

The safety system performance should match the level of risk. In other words, the greater the level of process risk, the better the safety system one needs in order to control the risk. This requires identifying the individual risks and assessing their impact.

The most difficult step in the overall process for most organizations seems to be determining the required SIL (safety integrity level). This is not a direct measure of process risk, but rather a measure of the safety system performance required in order to control the risks identified earlier to an acceptable level. The standards describe qualitative methods on how this can be done. One method is outlined in Fig. 2. A potential problem with the qualitative methods is that they are subjective and very often not repeatable. Different organizations may review the same process and each come up with different SIL requirements. There are also quantitative methods available for determining the SIL. A problem for many trying to use quantitative methods, however, is that they must decide on a quantitative target safety goal. Deciding “what is a tolerable death rate” is something few people wish to do, and fewer companies wish to put in writing. One can just imagine an attorney saying, “What do you *mean* you considered it *tolerable* to *kill* four people every 100 million man hours?!”

## Develop Safety Requirements Specification

The next step consists of developing the safety requirements specification, essentially the functional logic of the system. This will naturally vary for each system. There is no general, across-the-board recommendation that can be made. For example, if temperature sensor TT2301 exceeds 410°C, then close valves XV5301 and XV5302. Each safety function should have an associated SIL requirement, as well as any reliability requirements if nuisance trips are a concern. One should include *all* operating conditions of the process, from start-up through shutdown, as well as maintenance. (One may find that certain logic conditions conflict during different operating modes of the process.)



**FIGURE 2** One qualitative method of determining the required SIL.

The system will be programmed and tested according to the logic determined during this step. If an error is made here, it will carry through for the rest of the design. It won't matter how redundant or how often the system is manually tested—it will not work properly when required. These are referred to as systematic, or functional, failures. Using diverse redundant systems, programmed by different people using different languages and tested by an independent team, will not help in this situation, because the functional logic they all based their work on could be in error.

### Conceptual SIS Design

One doesn't pick a certain size jet engine for an aircraft based on intuition. One doesn't size a million dollar compressor by gut feel. One doesn't determine the size of pilings required for a bridge by trail and error (at least not any more).

The purpose of this step is to develop an initial design in order to see if it meets the safety requirements and SIL performance requirements. Initially one needs to select a technology, configuration (architecture), test interval, and so on. This pertains to the field devices as well as the logic box. Factors to consider are overall size, budget, complexity, speed of response, communication requirements, interface requirements, method of implementing bypasses, testing, and so on. One can then perform a relatively simple quantitative analysis to see if the proposed system meets the performance requirements [9]–[12], or make a qualitative judgment based on prior experience (although this is obviously harder to substantiate). The intent is to evaluate the system *before* one specifies the solution. Just as it is better to perform a HAZOP *before* you build the plant (it's hard to change the design once it's already been built), it is better to analyze the proposed safety system *before* you specify it, or else how will you know if it meets the performance goal?

### Detailed SIS Design

The purpose of this step is to finalize and document the design. Once a design has been chosen, the system must be engineered and built following strict and conservative procedures. This is the only realistic method of preventing design and implementation errors that we know of. The process requires

thorough documentation, that is, an auditable trail that someone else may follow for verification purposes.

### **Installation and Commissioning**

This step is to ensure the system is installed per the design and performs per the safety requirements specification. Before a system is shipped from a factory, it must be thoroughly tested for proper operation. If any changes are required, they should be made at the factory, not at the installation site. At installation, the entire system, this time including the field devices, must be checked as well. There should be a detailed installation document outlining each procedure to be carried out. Finished operations should be signed off in writing, showing that each function and operational step has been checked.

### **Operations and Maintenance**

In order to function properly, every system requires periodic maintenance. Not all faults are self-revealing, so *every* safety system *must* be periodically tested in order to make sure it will respond properly to an actual demand. The frequency of inspection and testing will have been determined earlier in the life cycle. All testing must be documented.

### **Modifications**

As process conditions change, it will be necessary to make changes to the safety system. All proposed changes require returning to the appropriate phase of the life-cycle. A change that may be considered minor by one individual may actually have a major impact to the overall process. This can only be realized if the change is thoroughly reviewed by a qualified team. Hindsight has shown that many accidents have been caused by this lack of review [13].

### **Decommissioning**

Decommissioning a system should entail a review to make sure removing the system from service will not impact the process or surrounding units, and that means are available during the decommissioning process to protect the personnel, equipment, and environment.

## ***MULTIPLE INDEPENDENT SAFETY LAYERS***

---

Figure 3 appears in a number of different formats in most all of the standards. It shows how there are various safety layers, some of which are prevention layers, others which are mitigation layers. The basic concept is simple: don't put all your eggs in one basket. (Everything fails; it's just a matter of when.) The more layers there are, the better. In addition, each layer should be as simple as possible, and the failure of one layer should not prevent another layer from performing its intended function. Some refer to this as defense in depth.

### **Process Plant Design**

The process plant itself must be designed with safety in mind. This is why HAZOP (Hazard and Operability Studies) and other reviews are performed, such as fault trees, checklists, what-if, and so on.

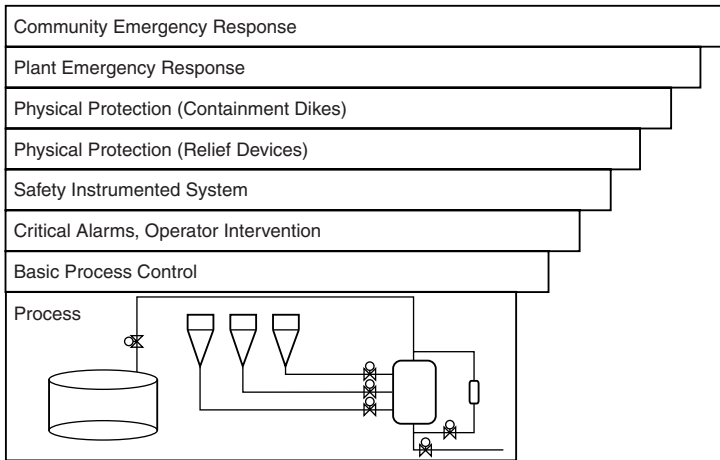


FIGURE 3 Multiple independent safety layers.

A major thrust within the process industry is to design inherently safe plants. Don't design a dangerous plant with the intention of throwing on lots of band aids to fix the problem. Design it so the band aids aren't even necessary. Work with low-pressure designs, low inventories, nonhazardous materials, and so on. Eliminating or reducing hazards often results in a simpler design, which may, in itself, reduce risk. The alternative is to add protective equipment to control hazards, which usually adds complexity.

## Process Control System

The process control system is the next layer of safety. It controls the plant for optimum fuel usage, product quality, and so on, and it keeps all variables (e.g., pressure, temperature, level, and flow) within safe bounds. Some are reluctant to consider the process control system a safety layer, yet this author has no such problem, as long as it is not the *only* safety layer.

## Alarms and Operators

If the process control system fails to do its function (for any number of reasons), alarms may be used to alert the operators that some form of intervention is required on their part. Alarm and monitoring systems should (1) detect problems as soon as possible, at a level low enough to ensure action can be taken before hazardous states are reached, (2) be independent of the devices they are monitoring (they should not fail if the system they are monitoring fails), (3) add as little complexity as possible, and (4) be easy to maintain, check, and calibrate.

Alarm and monitoring systems constitute the safety layer at which people get actively involved. Operators will usually be required for the simple reason that not everything can be automated. It is essentially impossible for designers to anticipate every possible set of conditions that might occur. Human operators may need to be considered since only they will be flexible and adaptable enough in certain situations. This is a two-edged sword, however, because events not considered in the design stage will no doubt also not be included in operator training either. In contrast, simply blindly following procedures has resulted in accidents [14]–[16]. Deviation from the rules is a hallmark of

experienced people, but it is bound to lead to occasional human error and related blame after the fact.

### **Shutdown/Interlock Systems**

If the control system and the operators fail to act, automatic shutdown systems take control. These systems are usually completely separate, with their own final elements. These systems require a higher degree of security to prevent inadvertent changes and tampering, and a greater level of fault diagnostics. The focus of this chapter is on these systems.

### **Fire and Gas Systems**

If the shutdown system fails and an accident ensues, fire and gas systems may be used to mitigate or lessen the consequences of the event. In the U.S., these are traditionally alarm-only systems—they do not take any automatic control actions. Typically, the fire crews must go out and manually put out the fire. Outside the U.S., these systems frequently take some form of control actions, or they may be integrated with the shutdown system.

### **Containment Systems**

If an atmospheric storage tank were to burst, dikes could be available to contain the release. In nuclear power plants, reactor are usually housed in containment buildings to prevent accidental releases. (The Soviet reactor at Chernobyl did not have a containment building, whereas the U.S. reactor at Three Mile Island did.)

### **Evacuation Procedures**

In the event of a catastrophic release, evacuation procedures are used to evacuate plant personnel from the area, and if necessary, even the outside community. While these are procedures only, and not a physical system (apart from sirens), they may still be considered one of the overall safety layers.

## ***SYSTEM TECHNOLOGIES***

---

### **Logic Systems**

There are a number of technologies available for use in shutdown systems—pneumatic, electromechanical relays, solid state, and PLCs (programmable logic controllers). There is no one overall best system; each has advantages and disadvantages. The decision of which system may be best suited for an application will depend upon many factors, such as budget, size, level of risk, flexibility, maintenance, interface and communication requirements, security, and so on.

Pneumatic systems are most suitable for small applications in which there are concern over simplicity, intrinsic safety, and lack of available electrical power.

Relay systems are fairly simple, are relatively inexpensive to purchase, are immune to most forms of EMI/RFI interference, and can be built for many different voltage ranges. They generally do not incorporate any form of interface or communications. Changes to logic require a manual change of documentation. In general, relay systems are usually only used for relatively small applications.

Solid-state systems (hardwired systems that do not incorporate software) are also available. Several of these systems were built specifically for safety applications and include features for testing,

bypasses, and communications. Logic changes still require a manual change of documentation. These systems have fallen out of favor with many people because of their high cost, along with the acceptance of software-based systems.

Software-based systems, generally industrial PLCs, offer software flexibility, self-documentation, communications, and higher level interfaces. Unfortunately, many general purpose systems were not designed specifically for safety and do not offer features required for more critical applications (such as effective self-diagnostics). However, certain specialized dual and triplicated systems were developed for more critical applications and have become firmly established in the process industry.

## Field Devices

In the process industries more hardware faults occur in the peripheral equipment—that is, the measuring instruments/transmitters and the control valves—than in the logic system itself. The overall reliability of a computerized control system may therefore not be significantly different than a conventional hardwired electrical system [17].

## Sensors

Sensors are used to measure process variables, such as temperature, pressure, flow, level, and so on. They may consist of simple pneumatic or electric switches, which change state when a setpoint is reached, or they may contain pneumatic or electric analog transmitters, which give a variable output in relation to the strength or level of the process variable.

Sensors, like any other device, may fail in a number of different manners. They may cause nuisance trips, that is, respond without any change of input signal. They may also fail to respond to an actual change of input conditions. While these are the two failure modes of most concern for safety systems, there are additional failure modes as well, such as leaking, erratic output, and responding at an incorrect level.

Most shutdown systems are designed to be fail safe. This usually means that when power is lost, the safety system makes the process revert to a safe state, which usually means stopping production. (Nuisance trips should be avoided for safety reasons as well. Startup and shutdown modes of operation involve the highest levels of risk.) Thought must be given to how the sensors should respond in order to be fail-safe. This usually means the sensor has normally closed and energized contacts, although this is not always the case. Transmitters can usually be configured to fail upscale or downscale in the event of a failure of the internal electronics. Thought should be given to the failure mode for each type of transmitter. A recommendation that is overall, across the board, and the same for all applications simply cannot be made.

Some measurements may be inferred from other variables. For example, if a system is designed to shutdown as a result of high pressure, it may be helpful to monitor temperature (if, because of the process, an elevated temperature might also imply a high pressure). Special care should be taken when operating sensors at the low end of their ranges, because of the potential low-accuracy problems. For example, a sensor designed to operate at 1,000 psi may not be able to differentiate between 20 and 25 psi.

**Technologies.** Discrete switches do not provide any form of diagnostic information. For example, if under normal circumstances a pressure sensor has a closed set of alarm contacts that are designed to open upon low pressure, and the contacts become stuck and cannot open as a result, the sensor has no means to annunciate the failure. The only way to know whether devices such as these are working is to *test* them.

Transmitters, in contrast, provide an analog signal in relation to the input variable, thus indicating in a limited sense whether the device is functioning properly. Any information is better than none. However, if the transmitter output is never monitored by the operators, or if the logic system does not

automatically check for “noise” or occasional drifting of the signal, then there really may be no more usable information than a discrete switch. It would be like having a color printer, but only printing black and white documents. The perceived benefit of having the color printer is illusory if one is unable to take advantage of the desired feature.

**Redundancy.** If the failure of any one sensor is of concern (i.e., a nuisance trip or a fail-to-function failure), then redundant or multiple sensors may be used. Ideally, the possibility of two sensors failing at the same time should be very remote. Unfortunately, this does not account for common cause failures, which might impact multiple sensors at the same time. Common cause failures are usually associated with external environmental factors such as heat, vibration, corrosion, and plugging. If multiple sensors are to be used, they should be connected to the process using different taps, so as to avoid common cause plugging failures. Consideration may be given to using different sensors from different manufacturers, or having different maintenance personnel work on the sensors (so as to avoid the possibility of a maintenance technician incorrectly calibrating all the sensors).

## Final Elements

Final elements generally have the highest failure rates. They are mechanical devices and subject to harsh process conditions. Safety shutoff valves also suffer from the fact that they are usually open and not activated for long periods of time, except for testing. One of the most common failure modes is that the valve is stuck, or frozen in place.

Valves should be fail safe upon loss of power. This usually requires a spring. A pneumatic or hydraulic valve would require a volume bottle to be fail safe, but the “availability” of the bottles may be too poor to rely on.

**Solenoids.** Solenoids are one of the most critical components of final elements. It is important to use a good industrial grade solenoid valve, especially for outdoor use. The valve must be able to withstand high temperatures, including the heat generated by the coil itself. In general, the reliability of solenoids is very low. One of the most common failures is burning out a coil, which causes a false trip. A dual coil would keep the solenoid energized if one coil were to burn out. Solenoids should be tested frequently.

## SYSTEM ANALYSIS

---

What is suitable for SIL 1, for SIL 2, and SIL 3? Things are *not* as intuitively obvious as they may seem. Dual is *not* always better than simplex, and triple is *not* always better than dual. Which technology to use, what level of redundancy, what manual test interval, and what about the field devices?

We do not design nuclear power plants or aircraft by gut feel or intuition. As engineers, we must rely on quantitative evaluations as the basis for our judgments. A quantitative analysis may be imprecise and imperfect, but it nevertheless is a valuable exercise for the following reasons:

1. It provides an early indication of a system’s potential to meet the design requirements.
2. It enables one to determine the weak link in the system (and fix it, if necessary).

In order to predict the performance of a system, one needs performance data of all the components. Information is available from user records, vendor records, military-style predictions, and commercially available databases in different industries.

When modeling the performance of a safety system, one needs to consider two failure modes. Safe failures result in nuisance trips and lost production. The preferred term for these failures is the nuisance trip rate (measured in years). Dangerous failures result in hidden failures in which the system

will not respond when required. Common terms used to quantify performance in this mode are pfd (probability of failure on demand), RRF (Risk Reduction Factor), and SA (Safety Availability).

There are a number of modeling methods used to predict safety system performance. The ISA technical report TR84.02 [9] provides an overview of simplified algebraic equations, fault trees, and Markov models. Each method has its pros and cons. No method is more right or wrong than any other, as they are all simplifications and account for different factors. Using such techniques, one can model different technologies, levels of redundancy, test intervals, and field device configurations. One can model systems using a hand calculator or develop spreadsheets or stand-alone programs to automate and simplify the task.

**TABLE 1** General System Recommendations

SIL	Sensors	Subsystem Logic	Final Elements
1	simplex switches	relays solid-state systems general purpose PLCs	simplex dumb
2	redundant switches simplex transmitters	relays fail-safe or fully tested solid-state systems certified software-based systems (simplex or redundant)	redundant dumb simplex smart
3	redundant switches redundant transmitters	relays fail-safe or fully tested solid-state systems certified software-based systems (dual or triplicated)	redundant smart

**Notes for Table 1**

Such tables are by their very nature oversimplifications. It is not possible to show the impact of *all* design features (failure rates, failure mode splits, diagnostic levels, quantities, manual test intervals, common cause factors, etc.) in a single table. Users are urged to perform their own analyses in order to justify their design decisions. The above table should be considered an example only, based on the following assumptions:

1. Separate logic systems are assumed for safety applications. Safety functions should not be performed solely within the BPCS (basic process control system).
2. Field devices are assumed to have an MTBF in both failure modes (safe and dangerous) of 100 years.
3. Simplex transmitters are assumed to have 80% diagnostics; redundant transmitters >95%.
4. Dumb valves offer no self-diagnostics; smart valves are assumed to offer 80% diagnostics.
5. When a consideration is made of solid-state logic systems, only solid-state systems specifically build for safety applications should be considered. These systems are either inherently fail safe (like relays) or offer extensive self-diagnostics.
6. General purpose PLCs are only appropriate for the lowest safety levels. They do not offer effective enough diagnostic levels to meet the higher performance requirements. Check with your vendors for further details.
7. One year manual testing is assumed for all devices. (More frequent testing would offer higher levels of safety performance.)

8. Redundant configurations are assumed to be either 1oo2 or 2oo3. 1oo2 configurations are safe, at the expense of more nuisance trips. 2oo2 configurations are less safe than simplex and should only be used if it can be documented that they meet the overall safety requirements.
9. The above table does not categorize the nuisance trip performance of any of the systems.

## KEY POINTS

---

- follow the steps defined in the safety design life cycle
- if you can't define it, you can't control it
- justify and *document* all of your decisions (i.e., leave an auditable trail)
- the goal is to have an inherently safe process (i.e., one where you don't even need an SIS)
- don't put all of your eggs in one basket (i.e., have multiple, independent safety layers)
- the SIS should be fail safe and/or fault tolerant
- analyze the problem *before* you specify the solution
- all systems *must* be periodically tested
- *never* leave points in bypass during normal operation (or be prepared to suffer the consequences)

## RULES OF THUMB

---

- maximize diagnostics (This is the most critical factor in safety performance.)
- any indication is better than none (e.g., transmitters have advantages over switches, systems should provide indications even signals are in bypass, etc.)
- minimize potential common cause problems
- general purpose PLCs are not suitable for the higher safety integrity levels
- when possible, use independently approved and/or certified components/systems (e.g., FM, TÜV, etc.)

## REFERENCES

---

1. "Application of Safety Instrumented Systems for the Process Industries," ANSI/ISA-S84.01-, 1996.
2. "Functional Safety—Safety Related Systems," IEC Draft Standard 61508, 1997.
3. *Guidelines for Safe Automation of Chemical Processes*, AIChE, CCPS, 1993.
4. CFR Part 1910.119, Process Safety Management of Highly Hazardous Chemicals U.S. Federal Register, February 24, 1992.
5. Leveson, N. G.; *Safeware—System Safety and Computers*, Addison-Wesley, Reading, Mass., 1995.
6. *Guidelines for Hazard Evaluation Procedures*, AIChE CCPS, 1992.
7. *Guidelines for Chemical Process Quantitative Risk Analysis*, AIChE CCPS, 1989.
8. Taylor, J. R., *Risk Analysis for Process Plants, Pipelines and Transport*, E & FN Spon, an Imprint of Chapman & Hall, London, UK, 1994.
9. "Safety Instrumented System (SIS)—Safety Integrity Level (SIL) Evaluation Techniques," ISA Draft Tec. Rep. dTR84.02, 1997.
10. Gruhn, P., "The Evaluation of Safety Instrumented Systems—Tools to Peer Past the Hype," *ISA transactions* Vol. 35, pp. 25–32, 1996.

11. Gruhn, P., "Safety Systems: Where is Your Weak Link?" *InTech*, December 1993.
12. Smith, D. J., *Reliability, Maintainability and Risk*, Butterworth Heinemann, Oxford, UK, 1993.
13. "Out of Control: Why Control Systems Go Wrong and How To Prevent Failure," *Health & Safety Executive* (UK), 1995.
14. Kletz, T. A., *What Went Wrong? Case Histories of Process Plant Disasters*, Gulf Publishing, Houston, TX, 1986.
15. Kletz, T. A., *An Engineer's View Of Human Error*, The Institute of Chemical Engineers, Warwickshire, England, 1985.
16. Kletz, T. A., *Lessons From Disaster—How Organizations Have No Memory and Accidents Recur*, Gulf Publishing, Houston, TX, 1993.
17. Lowe, X., *Measurement and Control*, Vol. 17; p. 317, 1984.

---

## AN OVERVIEW OF THE ISA/IEC FIELDBUS

by Terry Blevins\*

---

### INTRODUCTION

The Instrument Society of America (ISA) approved in 1985 the charter of the SP50 committee to develop a digital fieldbus standard. This standard defines a digital, two way, multidrop communication link among intelligent field devices and automation systems. It provides for communications at 31.25 K baud over existing wiring and meets intrinsic safety requirements of the process industry. By being able to multidrop devices off one single pair of wires as shown in Fig. 1, a significant saving in wiring and termination cost of 30–40% can be achieved over a traditional installation. In addition, since devices communicate digitally over the fieldbus, it is possible to obtain or send multiple pieces of information to a fieldbus device, that is, the device is no longer restricted to providing or receiving a single measure or output value. To take maximum advantage of this capability, each fieldbus device supports a standard function block application. Through this function block application, it is possible for fieldbus devices to be used to meet application requirements for measurement, alarm, calculations, and control.

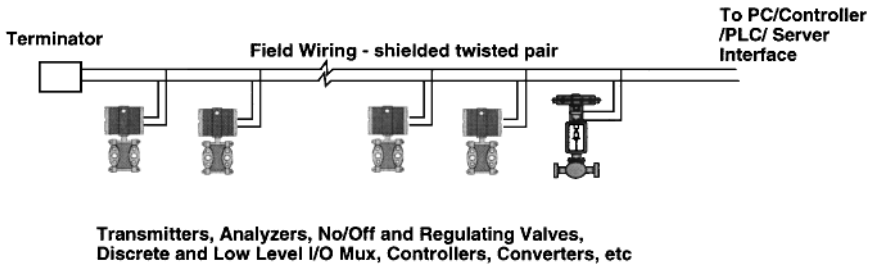
Within the International Electrotechnical Commission (IEC), a working group, SC65C WG6, is responsible for the international fieldbus standard. This working group holds joint meetings with SP50 on the physical layer and communication stack. A separate working group, TC65 WG6, is defining function block architecture, IEC1499, targeted to meet the requirements of both manufacturing and process automation.

The fieldbus standard defined by ISA/IEC ushers in the next generation of control and automation products and systems that represent change and opportunity for the process industry:

- advanced function added to field instruments
- expanded view for the operator

---

\* Fisher-Rosemount Systems, Inc. Austin, Texas.



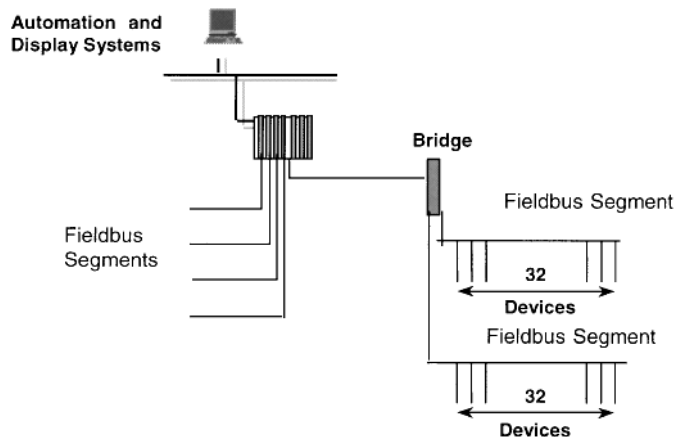
**FIGURE 1** Fieldbus segment: multiple devices may be connected to a single fieldbus segment.

- reduced wiring and installation costs
- reduced I/O equipment by one-half or more
- provide increased information flow to enable automation of engineering, maintenance, and support functions

The Fieldbus Foundation (FF) is an independent, nonprofit organization established to support commercialization of the ISA/IEC fieldbus standard. Foundation Fieldbus devices use the IEC physical layer and communication stack. The Function Block Application Process defined by the Fieldbus Foundation specification includes architectural concepts and terminology of IEC1499. Training schools are sponsored by the Foundation to support fieldbus device development. In addition, the Foundation has established test procedures, which are used to certify field devices as interoperable.

## PHYSICAL INSTALLATION OF A FIELDBUS SYSTEM

The normal twisted pair with shield wiring used in most plants today may serve as the fieldbus. Thus, in an existing installation, wiring may be reused when Foundation Fieldbus devices are installed. Typically, the fieldbus segments will be connected into a control system through a controller or control system bridge, which is specifically designed to support fieldbus devices, as illustrated in Fig. 2.



**FIGURE 2** System connection: fieldbus segments may be integrated into a control system by using a controller or bridge.

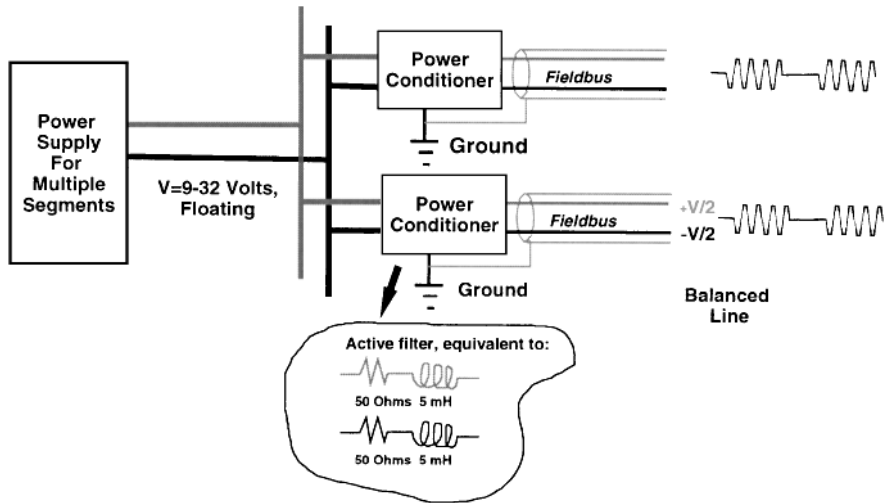


FIGURE 3 Segment Power: some power conditioners may be used with a common bulk power supply.

A maximum of 32 field devices may be connected to a single fieldbus segment. Twelve of the devices on a fieldbus segment may be powered from the fieldbus segment, similar to the powering of tradition two-wire transmitters. The power for each fieldbus segment is provided through a power conditioner. The purpose of the power conditioner is to prevent the communication signal from being attenuated by the power supply and to eliminate cross talk between fieldbus segments through a common power supply. Also, the power conditioner provides a balanced conductor for communications; that is, the fieldbus conductors are maintained at  $\pm 1/2$  the supply voltage with respect to ground. In some cases, a common power supply with floating output may be used to power multiple power conditions, as shown in Fig. 3.

In an intrinsically safe installation, power conditions with built-in barriers are available that support 4–6 fieldbus powered device. An example of an installation that utilizes commercially available power conditions is shown in Fig. 4.

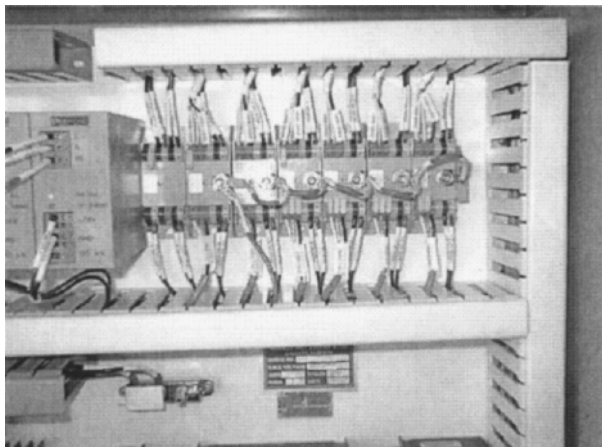
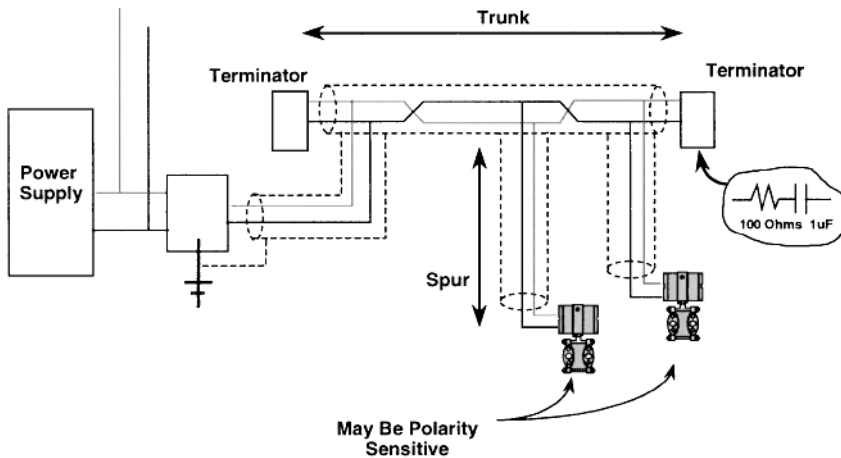


FIGURE 4 Example of a commercial power conditioner.



**FIGURE 5** Fieldbus Trunk: the main path of a fieldbus segment may have multiple spurs.

The main path of a fieldbus segment is commonly called the truck or home run, as illustrated in Fig. 5. Terminators are required at the far ends of the fieldbus trunk. Branches off the fieldbus trunk, known as spur, may be a maximum of 120 m in length. The total length of the trunk and spurs that make up a fieldbus segment may be up to 1,900 m when an 18-gage single twisted pair with shield wiring is used.

The maximum length of a fieldbus segment will depend on the type of cable used—as detailed in Table 1. Also, the maximum length of a spur will vary with the number of devices on the segment and the spur, as shown in Table 2.

**TABLE 1** Maximum Fieldbus Segment Length

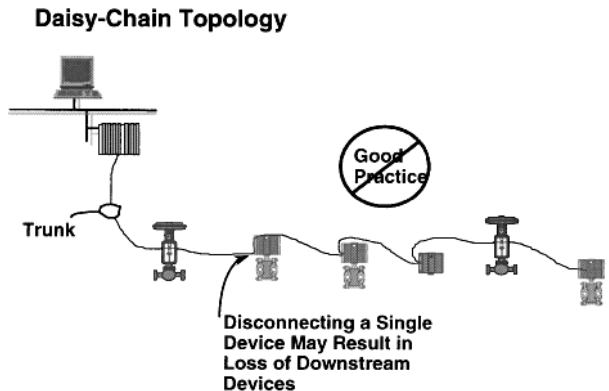
Type	Cable Description	Size (mm <sup>2</sup> )	Max Length <sup>a</sup> (m)
Type A	Shielded, twisted pair	0.8 (#18AWG)	1,900
Type B	Multitwisted pair, w/shield	0.32 (#22AWG)	1,200
Type C	Multitwisted pair, w/o shield	0.13 (# 26AWG)	400
Type D	Multicore, w/o shield	1.25 (#16AWG)	200

<sup>a</sup> Total of trunk plus spur(s) length, based on IEC-1158-2 and ISA S50.02-1192.

**TABLE 2** Maximum Recommended Fieldbus Spur Length<sup>a</sup>

Total devices	Device per Spur		
	1 Device (m)	2 Devices (m)	3 Devices (m)
1–12	120	90	60
13–14	90	60	30
15–18	60	30	1

<sup>a</sup> Based on IEC-1158-2 and ISA S50.02-1192 Part 2, Annex C (informative) notes: these lengths are “recommended” and are not required.

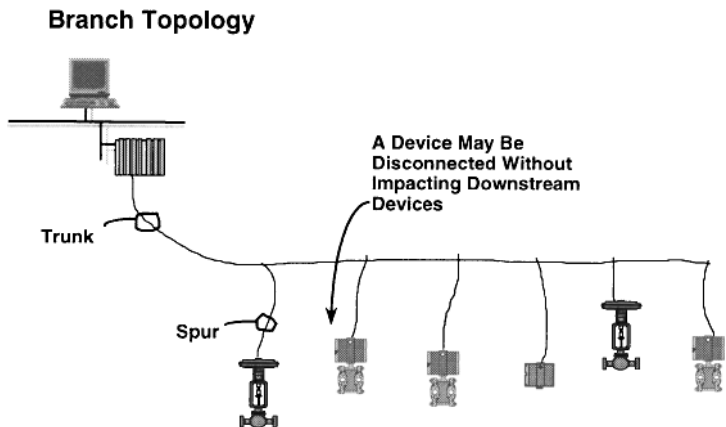


**FIGURE 6** Daisy-chain arrangement: it is not a good practice to daisy-chain fieldbus devices on a segment.

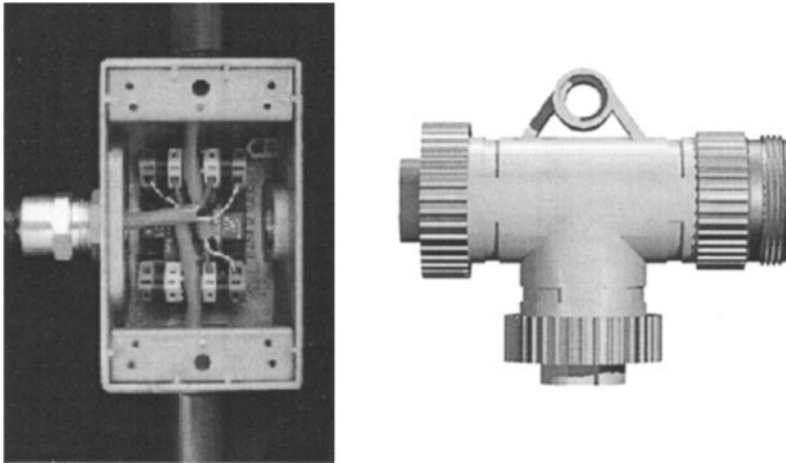
In the initial design of a fieldbus segment, one of the critical decisions is what devices will be placed on an individual fieldbus segment. When information is to be exchanged between fieldbus devices for control or calculations, then these devices should be located on the same segment. Naturally, there is a tendency to place as many devices on a fieldbus segment as possible to reduce the overall cost in wiring and fieldbus interface cards. However, depending on the speed with which information must be accessed, some manufacturers will limit the number of devices that may be placed on an individual segment.

Once the decision is made concerning what devices should be on a segment, then a detailed layout of the segment is possible. In the design of the segment, a key decision will be how the devices will be multidropped from the fieldbus segment. Even though it is possible to wire the fieldbus devices in a daisy-chain topology as shown in Fig. 6, this arrangement is not recommended because of the problems this may present in maintenance.

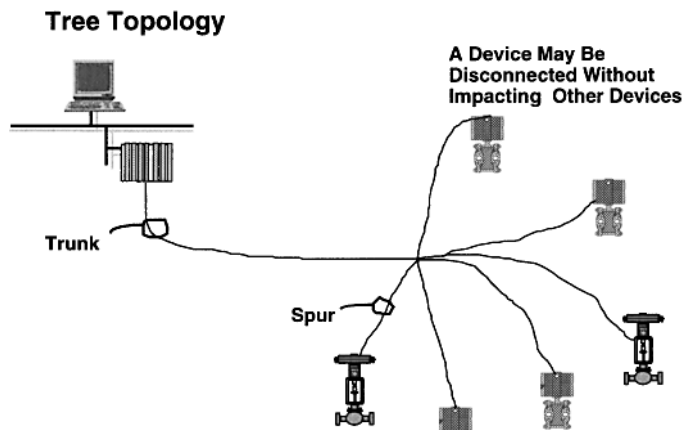
For example, in a daisy-chain arrangement, all power may be lost downstream of a fieldbus device that is removed for maintenance. When individual devices are to be multidropped off the fieldbus segment, then a branch topology, shown in Fig. 7, supports ease in maintenance, checkout, and installation.



**FIGURE 7** Branch topology: this allows a single device to be added or removed without disrupting other devices on the segment.



**FIGURE 8** Spur wiring: example using a conduit box and prefabricated T.

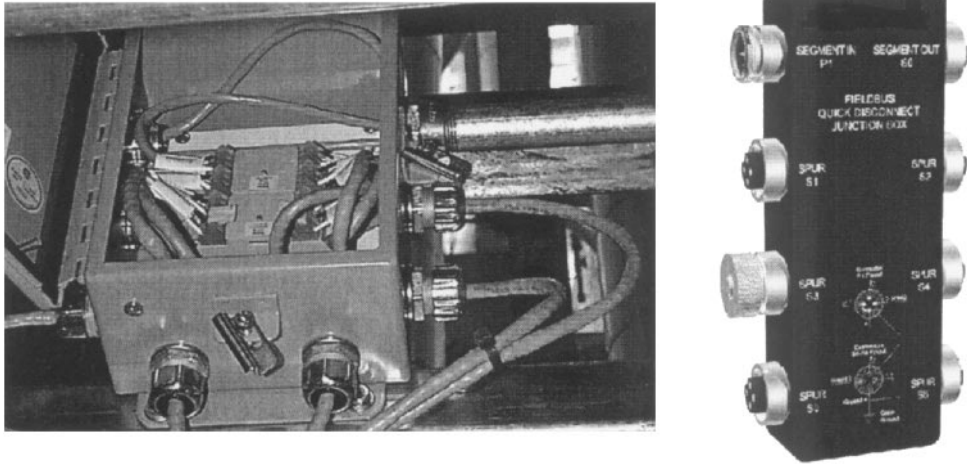


**FIGURE 9** Most fieldbus installations will utilize a tree topology.

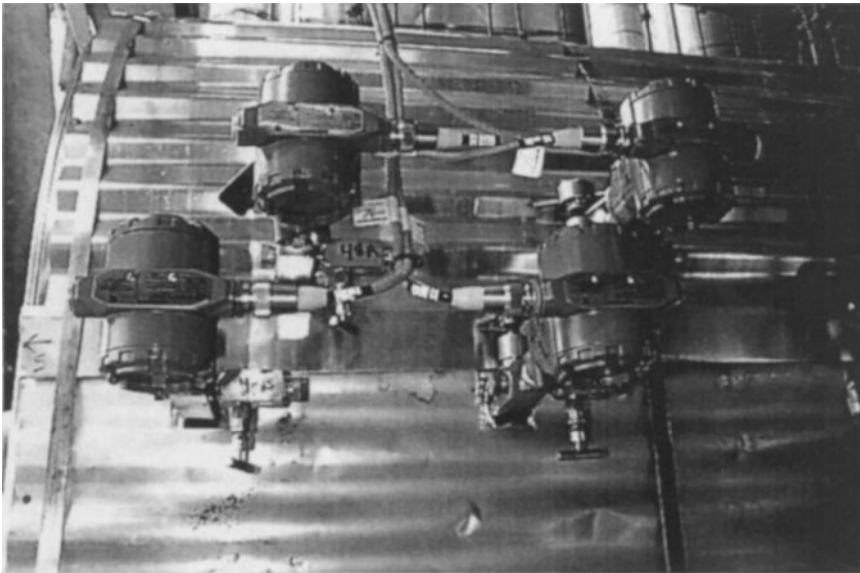
When the branch topology is used, it is possible to add or remove individual devices from the fieldbus segment without disruption to other devices on the segment. A number of commercial products are available to support the installation of the branch topology. When conduit and junction boxes are selected, a commercially available junction board may be installed in the conduit box to facilitate wiring between the main segment and the spur. If quick disconnects are installed, then commercially available T's may be used for the spur connection to the fieldbus trunk. Examples of spur wiring using a conduit and T are shown in Fig. 8.

In a typical installation, multiple fieldbus devices may be located in the same physical area. In this case, the spurs from these devices may be wired together in a junction box to form a tree topology, as shown in Fig. 9.

Traditional terminal strips may be used for wiring inside the junction box. Alternately, commercial DIN rail-mounted termination blocks are available for fieldbus installation. Using such devices will minimize the potential for wiring mistakes. These termination blocks are available with a built-in



**FIGURE 10** Fieldbus junction box: pre-fabricated components may reduce installation and checkout time.



**FIGURE 11** Example installation in which the field devices were purchased with quick disconnects.

terminator for use at the far end of the segment. Also, prefabricated junction boxes are available for quick disconnect terminations, as shown in Fig. 10.

In some installations, as many as 16 fieldbus devices will be connected to a single fieldbus segment. In such cases, a short along the fieldbus segment could result in the loss of communications with all devices on the segment. To support maintenance, many manufacturers will provide quick disconnects at the fieldbus device to minimize the chance of shorting the segment while adding or removing a field device. Examples of fieldbus device quick disconnects are shown in Fig. 11.

## UTILIZING FIELDBUS DEVICES TO MEET APPLICATION REQUIREMENTS

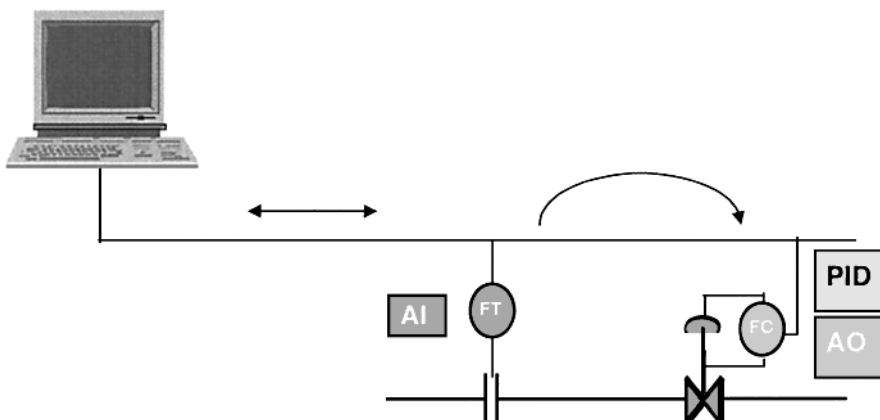
In a fieldbus environment, the user application is defined through the configuration of function blocks. This approach is similar to the configuration of control and monitoring in a distributed control system (DCS) today. However, the fieldbus function blocks support applications, which involve measurement and control, to be distributed between fieldbus devices. Such capability will allow the base level process control and measurement done today in distributed control systems and single-loop digital controllers to be implemented in a fieldbus device. Some of the advantages gained by standardizing the user application are:

- consistent, easy, block-oriented configuration of functions
- distribution and execution of function in field devices from different manufacturers in an integrated, seamless manner
- consistent definition of information that will be communicated and function that will be executed
- avoidance of custom interfaces and cumbersome mapping

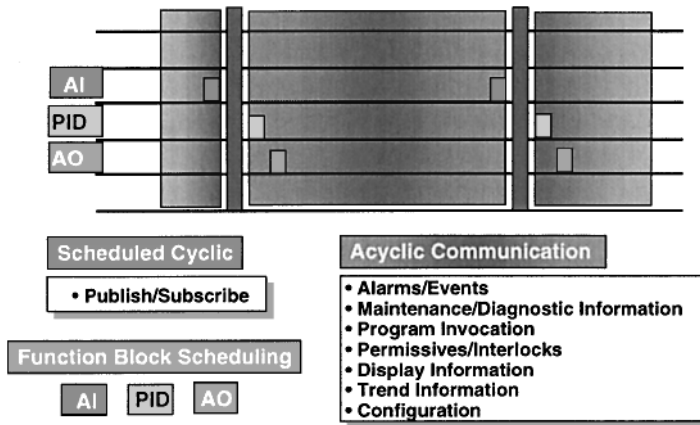
Monitoring, calculation, and control functions may be defined by configuring function blocks within a fieldbus device and configuring the connections between function block input and output parameters. Fieldbus devices provide a new level of capability. Fieldbus valve and transmitters may support control and calculations. In addition, auxiliary measurements such as stem position or limit switch status for an on-off valve will often be made available through fieldbus.

The function block application in Foundation Fieldbus devices is designed to allow control, measurement, and calculation functions to be distributed between field devices. For example, a flow measurement might be implemented in a transmitter using the analog input block, AI block. This transmitter publishes its measurement value and its status as shown in Fig. 12. A valve on the fieldbus segment may support a control block, the PID block, which subscribes to the published measurement value. Based on this value and its target setpoint, the PID block may calculate an output required to maintain the setpoint. Within the valve, this output might be used by an output block, the AO block, to adjust the valve.

To successfully distribute control between fieldbus devices without degrading control by communication delay, the scheduling of function block execution and related communications is critical. Thus, as an integral part of the Foundation Fieldbus specification, scheduling of communications in



**FIGURE 12** Fieldbus communication: support is provided to distribute control and calculations between fieldbus devices.



**FIGURE 13** Both control-related communication and function block execution may be scheduled to minimize delay.

conjunction with function block execution is defined. A typical schedule for the above example is illustrated in Fig. 13. When no control communications are scheduled, then other information for display, alarm notification, and so on may be communicated.

Function blocks and tools used for process analysis, such as trending of measurement versus time, often assume periodic sampled values. In such cases, it may not be practical to compensate for variation or jitter in the sample time. Thus, fieldbus allows measurements to be sampled on a precisely periodic basis, independent of fieldbus communications. Through the scheduling of block execution, these sampled values may be made available to control blocks with minimal delay.

The low processing power of intrinsically safe devices, combined with the distribution of control between fieldbus devices, imposes some constraints on the design of fieldbus devices. To support the periodic sampling of inputs, a common sense of time is maintained in each device on a fieldbus segment. Each device maintains a schedule of which blocks within the device are to be executed. Based on this, it is possible to schedule communication of block output values between devices for calculations and control.

To obtain an accurate measurement, a device may sample a process measurement at a much faster rate than is needed for control or monitoring requirements. For example, a measurement may be processed 20 times/s, even though it may be used in control only 5 times/s. A transducer block is defined in fieldbus devices to contain the parameters associated with the basic measurement. The processing of the transducer block is defined to be independent of function blocks that reference the transducer block, as shown in Fig. 14. To accommodate these differences in execution rates, software filtering is provided in transducer blocks. Through this filter, the frequency content of the measurement may be matched to the function block execution rate.

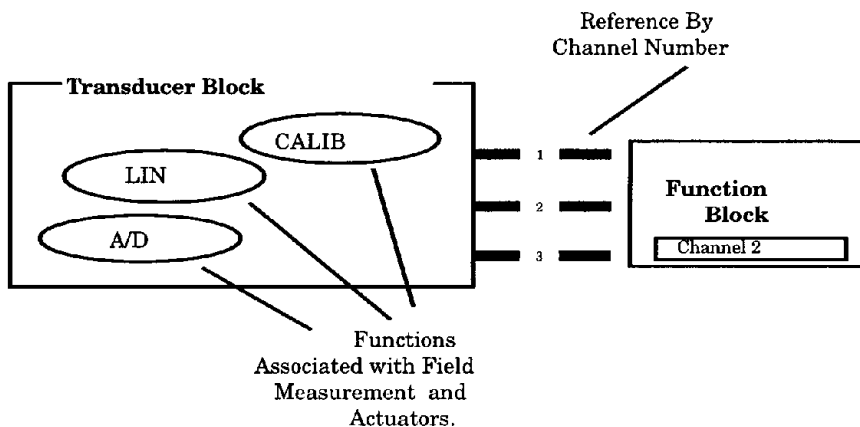
As part of input block processing, checks on associated hardware and software are performed. In addition, process alarm detection may be done. On the detection of a change in block status or process alarm condition, an event notification will be generated. Included in this notification will be the time of detection and a status that gives further information on the alarm or event.

Ten basic blocks and 19 advanced function blocks have been defined by the function block specification, as shown in Table 3. These blocks allow both analog and discrete monitoring, calculations, and control to be done in field devices. It is estimated that as much as 80% of the DCS and PLC controller functionality may be distributed to the fieldbus devices using this capability.

Using these function blocks, most analog and discrete control problems may be addressed. In Tables 4 and 5, some examples of how Foundation fieldbus function blocks may be applied in typical control applications are shown.

**TABLE 3** Function Blocks Defined by Foundation Fieldbus Specification

Basic	
discrete input (DI)	P, PD Control (PD)
discrete output (DO)	control selector (CS)
analog input (AI)	manual loader (ML)
analog output (AO)	bias/gain station (BG)
PID, PI, I control (PID)	ratio station (RA)
Advanced	
pulse input (PCI)	deadtime (DT)
complex analog output (CAO)	arithmetic (ARITH)
complex discrete output (CDO)	calculate (CALC)
step output PID (STEP)	integrator (INT)
device control (DC)	timer (TIM)
setpoint ramp generator (SPR)	analog alarm (AALM)
splitter (SPLIT)	discrete alarm (DALM)
input selector (ISEL)	analog human interface (AHI)
signal characterizer (CHAR)	discrete human interface (DHI)
lead lag (LL)	

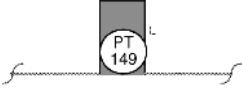
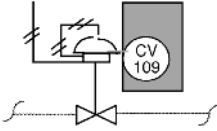
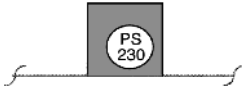
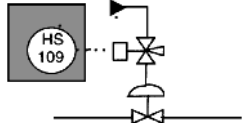
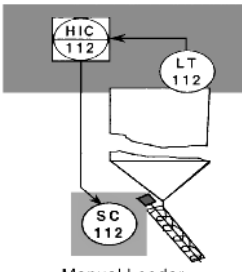
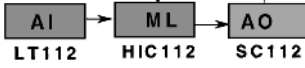
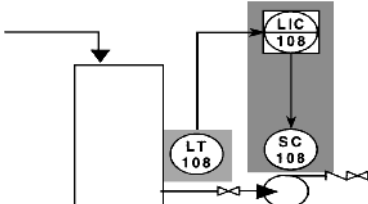
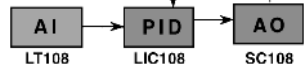
**FIGURE 14** Input and output function blocks may reference the field value(s) provided by the transducer block.

Before a field device is certified as interoperable by the foundation, it is put through a series of test to verify that the device conforms to the definitions of the specification. Only if the device supports the block features defined by the function block specification can it earn the interoperable check issue by the Fieldbus Foundation. In purchasing a fieldbus device, then it will be important to verify that the device has passed interoperability testing.

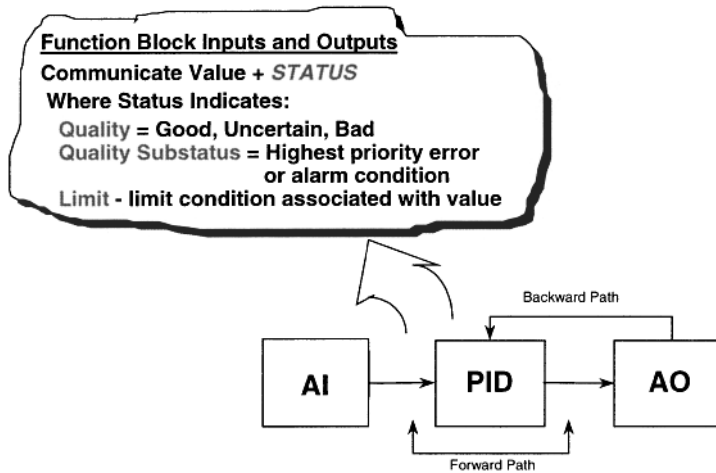
## **DIAGNOSTIC SUPPORT OF FOUNDATION FIELDBUS DEVICES**

The ability to quickly troubleshoot a process and its control and instrumentation system is an important aspect of plant operation. Any time spent identifying the source of a malfunction may contribute to

**TABLE 4** Example Applications Addressed by Fieldbus Devices

Control/Masurement Function	Fieldbus Function Blocks Required
 <p style="text-align: center;"><b>Transmitter</b></p>	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">AI</div>
 <p style="text-align: center;"><b>Regulating Valve</b></p>	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">AO</div>
 <p style="text-align: center;"><b>Pressure Switch</b></p>	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">DI</div>
 <p style="text-align: center;"><b>On/Off Valve</b></p>	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">DO</div>
 <p style="text-align: center;"><b>Manual Loader</b></p>	
 <p style="text-align: center;"><b>Single Loop Control</b></p>	

Control/Measurement Function	Fieldbus Function Blocks Required
<p data-bbox="236 174 532 198">Control/Measurement Function</p> <p data-bbox="274 530 532 554">Single Loop Control With Interlock</p>	
<p data-bbox="345 865 461 888">Ratio Control</p>	
<p data-bbox="341 1234 467 1258">Override Control</p>	
<p data-bbox="186 1525 343 1548">Control For Motors</p> <p data-bbox="309 1552 524 1575">Motor Control With Interlock</p>	



**FIGURE 15** Input/Output status: each function block input or output supports a value and status.

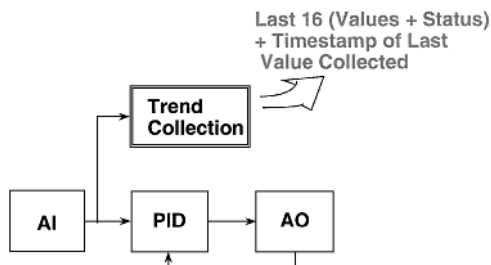
process downtime and lost production. In many digital control systems in use today, the process measurement is received as a 4–20 mA signal and the actuator is sent as a 4–20 mA signal. Plant instrument technicians are familiar with the troubleshooting process instrumented in this manner. Within fieldbus devices the signal from the measurement element will be converted directly to its digital representation. The only accessible representation of the measurement will be the digital value communicated over the fieldbus. Thus, the tools and techniques used in calibration, setup, commissioning, and troubleshooting will change. The design of any fieldbus system must address the requirements for process analysis through communicated values.

Features are included in the Fieldbus Foundation Specification for the support of process analysis and troubleshooting field problems. Special consideration is given to signal filtering and sampling to ensure that the resulting digital values adequately represent the true process measurement. Conditions detected by the field device that would impact the validity of the measurement are made available as the status of the digital value. To allow this information in a device to be used, an efficient mechanism is provided to access field device information without disrupting the control distributed between field devices.

As part of input block processing, checks on associated hardware and software are performed. Analog values will be transferred as floating point values in engineering units. The status of an output parameter is calculated by the block to give an explicit indication of the quality of the value: *good*, *uncertain*, or *bad* as shown in Fig. 15. A substatus attribute indicates the primary condition, which determines the quality. Also, there is an explicit indication in the status attribute of whether the value is limited high or low. When a block input or output parameter is access for viewing at an operator console, historian, or diagnostic tool, both the value and its associated status may be communicated. The quality information and its substatus may be useful in diagnosing the cause of a bad or uncertain measurement.

A substatus attribute of status indicates the primary condition that determines the quality. The substatus for good quality provides additional information that may be needed for control or monitoring purposes. For example, the good substatus for input blocks is defined to give further information on the status of alarms that are defined for the block. As part of block definition, the processing and propagation of the status is defined.

Communication support is provided for an explicit indication of stale data. Data are defined to be stale if the input value has not been updated since it was last read or if a value has been communicated but the value received is the last (old) value. Within a device, the input parameter status will be



**FIGURE 16** Diagnostic information may be obtained by using trend objects supported by a fieldbus device.

set to bad—no communication if the value is stale for a specified time. When communications are re-established, then the input will again reflect the status of the output parameter.

When a block input or output parameter is access for viewing at an operator console, historian, or diagnostic tool, both the value and its associated status may be communicated. The quality information and its substatus may be useful in diagnosing the cause of a bad or uncertain measurement. To facilitate the use of status information, the representation and meaning of quality and its associated substatus have been defined in the Foundation Fieldbus Specification.

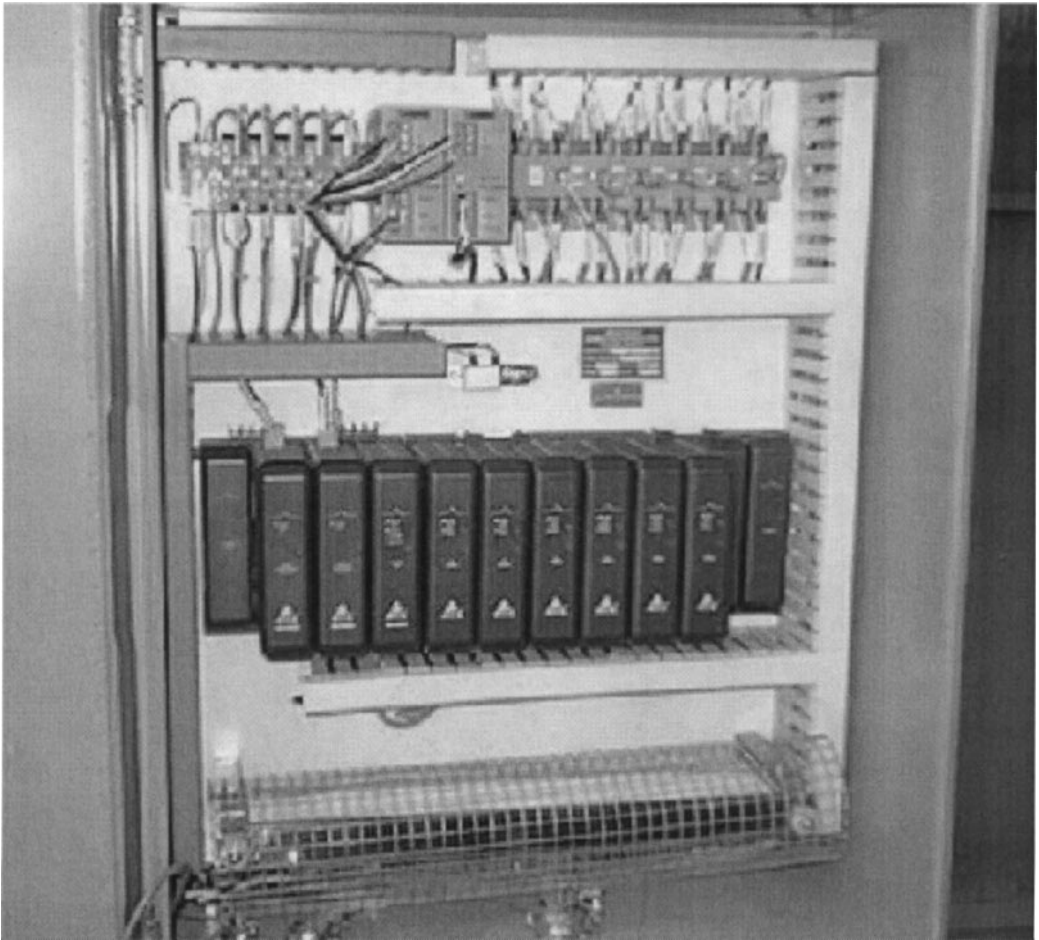
When information is accessed for process analysis and diagnostics, it may not be practical or technically feasible to communicate individual parameter values as fast as the block executes rate. To address these issues, trend objects are defined in fieldbus devices as illustrated in Fig. 16. At least one trend object is included in each field device. A trend object maintains a collection of 16 parameter samples. Through trend-object configuration, it is possible to specify the input or output parameter to be collected at the collection rate. The block collects both the parameter value and status. Samples rates are an integer multiple of the parameter's block execution rate. Sample values may be averaged if the collection rate is slower than the block execution rate.

When 16 new samples are collected, this information along with the time stamp of the last value collected will be automatically reported to one or more devices. Through this mechanism, the communication load associated with the collection of the parameter value and status may be reduced by as much as a factor of 16. In addition, by collecting samples at the device, the collection of information is not skewed or limited by the communication rate of the fieldbus.

View objects are provided for each function block to facilitate information access. A view object allows multiple parameters to be read or written with a single communication request. Through this mechanism, communications to support updating operator interfaces (etc.) may be done efficiently. The processing of such requests is done when no other communications are scheduled. In addition, process alarm detection may be done in the fieldbus device. On the detection of a change in block status or process alarm condition, an event notification will be generated. Included in this notification will be the time of detection and a substatus, which gives further information on the alarm or event.

## **CONTROL SYSTEM IMPACT**

A new generation of control systems that is designed to effectively utilize fieldbus devices is available from major control system vendors. Such systems are smaller in size physically because of the reduction in the number of I/O card required to interface with field devices. To allow the configuration and calculated parameter values in a fieldbus device to be accessed, each manufacturer will provide a device description (DD) of his or her fieldbus devices written in the device description language (DDL) standardized by the Fieldbus Foundation. Configuration and operator interface stations, which



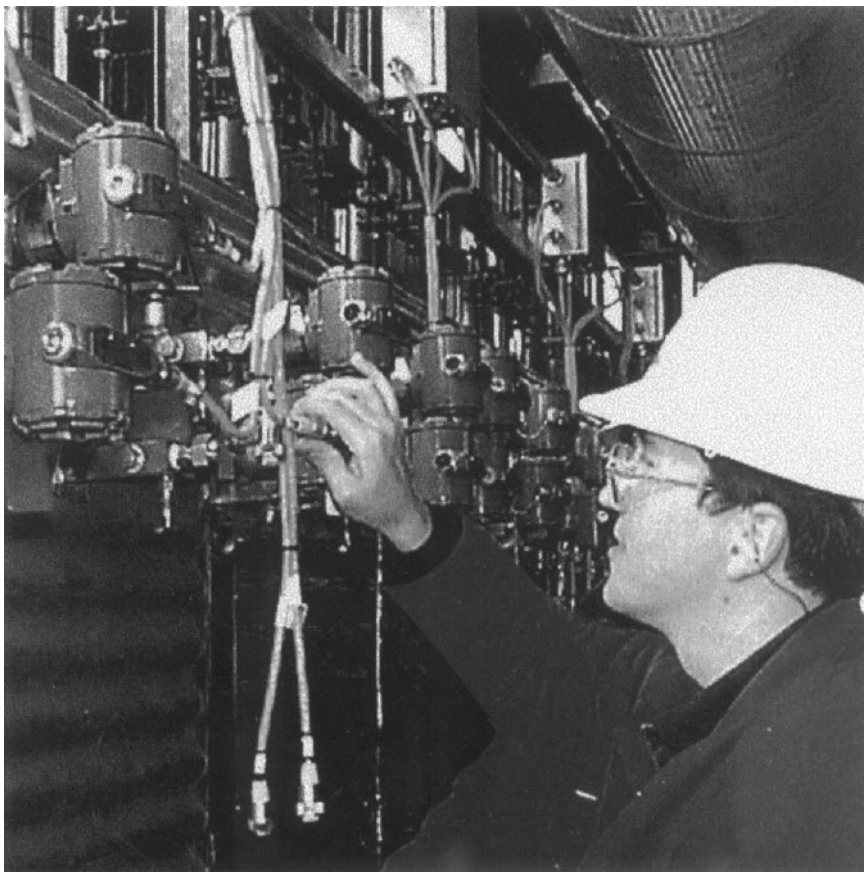
**FIGURE 17** Example installation of a fieldbus controller that is connected to 96 fieldbus devices.

have been designed to utilize device descriptions, will be able to access device function blocks after the device description is loaded. An example of a controller designed for fieldbus is shown in Fig. 17.

Information for process analysis and diagnostics is available in fieldbus devices that are based on the Fieldbus Foundation's specification. To take advantage of this capability, future tools for process analysis, diagnostics, and reporting will utilize more than the measurement values. In particular, these tools will access and make visible:

- alert notification time stamp and substatus detailing alarms or events
- quality and substatus accompanying function block inputs and outputs

To support the analysis of blocks executing at fast rates, future devices for viewing trends and archiving trend data will utilize value and status samples that are collected and reported through trend objects in the field devices.



**FIGURE 18** World's first large commercial installation based on the IEC/ISA fieldbus standard.

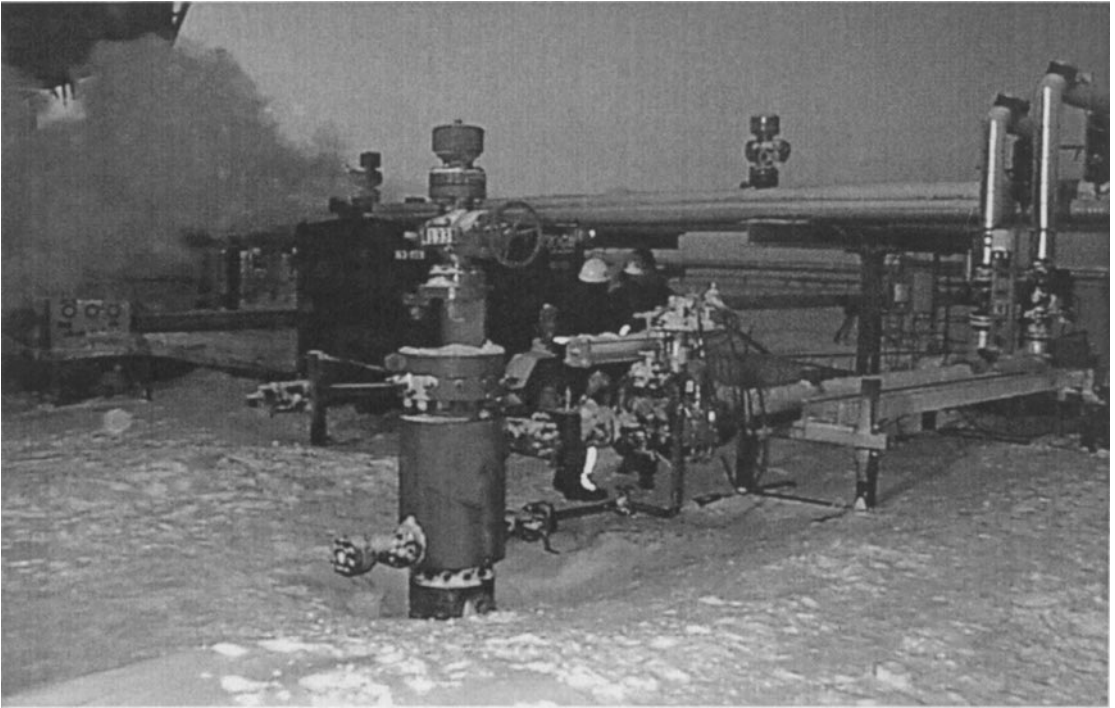
### ***EXAMPLE INSTALLATIONS: COMMERCIAL FIELDBUS INSTALLATIONS***

---

The first two major commercial fieldbus installations may be used to illustrate the labor and material savings that are achieved by using fieldbus. The world's first commercial installation of Foundation Fieldbus was at a Dow Chemical Company in Alberta, Canada. Mr. David Taylor was the project manager for this historic installation. Initially, a total of 96 fieldbus transmitters were installed at Dow in April, 1997 (shown in Fig. 18). Based on the success of this installation, an additional 480 fieldbus devices were installed in July, 1997.

At the Dow installation, quick disconnects and prefabricated fieldbus junction boxes are used extensively. Dow reported the total project cost as 30% lower than that documented for a similar installation that used traditional instrumentation.

The second major commercial Foundation Fieldbus installation was at ARCO, West Sak, Alaska. Mr. Duane Toavs was the ARCO project manager for the large installation. This project included fieldbus transmitters and valves for 30 oil wells (shown in Fig. 19). The system was commissioned in December, 1997.



**FIGURE 19** At the West Sak oil field on Alaska's North Slope, operated by ARCO Alaska, Inc.

ARCO reports the savings associated with the West Sak fieldbus installation as follows:

- 16% reduction in wellhead terminations
- 69% reduction in comparable wiring costs
- 98% reduction in home run wiring
- 83% reduction in instrument commissioning and checkout
- 90% reduced configuration time to add an expansion well
- 92% reduced engineering drawings to add or expand a well
- ~\$210,000 material cost savings
- ~\$320,000 labor savings
- ~\$90,000 engineering savings

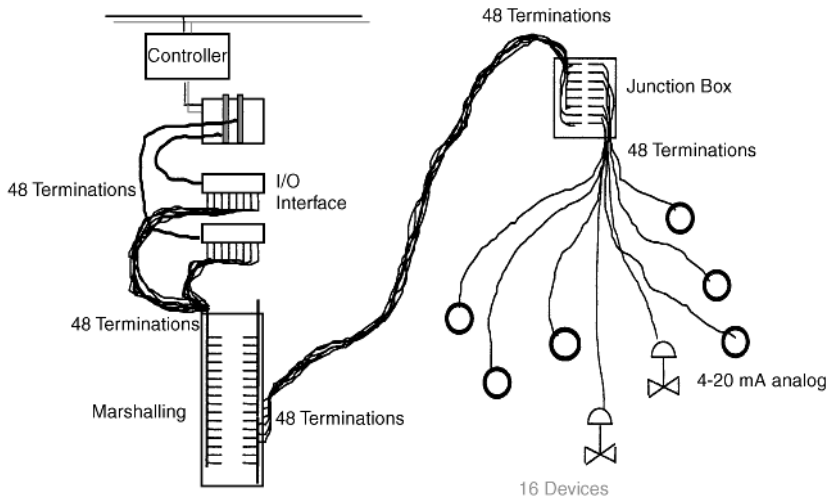
Additional long-term benefits are anticipated from the diagnostic and maintenance features of fieldbus devices.

### ***ESTIMATING SAVINGS FROM USING FIELDBUS TECHNOLOGY***

---

By the use of fieldbus technology, savings are possible in the following areas:

1. Reduction of terminations
2. Reduction in number of I/O cards



**FIGURE 20** Illustration of terminations for a traditional I/O with 16 devices.

3. Reduction in home run wiring
4. Reduction in instrument control room panel space

The magnitude of these cost savings may be determined by analyzing the controls configuration in a manufacturing plant. Each area of cost reduction may be addressed separately.

### Reduction of Terminations and Home Run Wiring

In a typical plant situation, each transmitter or valve is connected to a junction box with a single, shielded twisted pair, wire. Since each wire has three terminations, the number of terminations into the junction box is the number of devices multiplied by three (two conductors plus the shield). The junction box is often connected to a marshalling panel with the same number of terminations (number of devices times three), and finally, the marshalling panel is connected to the I/O cards with the same number of terminations. The schematic layout of the terminations in a traditional I/O system with 16 devices is shown in Fig. 20.

In a fieldbus situation, the number of terminations is significantly reduced. The same number of wires is required to connect the devices to the junction box, but one wire (three terminations) connects the junction box to the marshalling panel, and one wire (three terminations) connects the marshalling panel to the H1 fieldbus card, as illustrated in Fig. 21. Assuming the same number (16 devices) of valves and transmitters that are connected to the fieldbus segment, a maximum of 16 wires (48 terminations) can be connected to the fieldbus segment at the junction box, as shown in Table 6.

From a total cost perspective, a tremendous installation cost savings can be achieved with the usage of fieldbus technology. The reduction of 240 terminations to 60 for an installation of 16 devices represents a 75% reduction in termination cost. A significant savings in wiring reduction will result from any fieldbus installation. Using traditional I/O, 16 devices require 16 wires from the junction box to the marshalling panel and 16 wires from the marshalling panel to the I/O card. With fieldbus technology, one wire connects the junction box to the marshalling panel, and one wire connects the marshalling panel to the I/O card. As with a traditional I/O layout of 16 devices, a fieldbus layout of 16 devices requires 16 wires to the junction box. The reduction of 16 wires to 1 wire results in an 83% reduction.

**TABLE 6** Wiring Terminations for 16 Devices

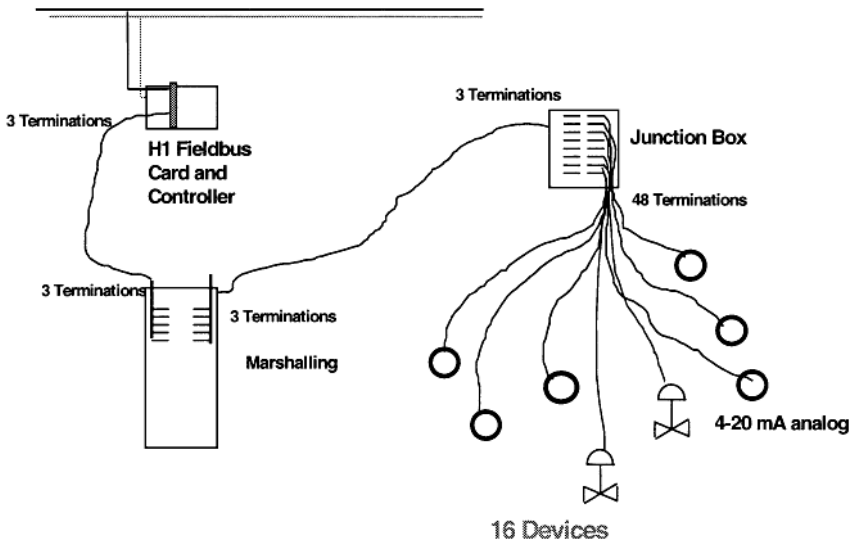
Terminations from the . . .	To the . . .	Traditional I/O Terminations	Fieldbus Terminations
16 devices (16 wires, times 3)	junction box	48	48
Junction box	marshalling panel	96	6
Marshalling panel	I/O cards	96	6
Total		240	60

**TABLE 7** Card Requirements for Traditional I/O

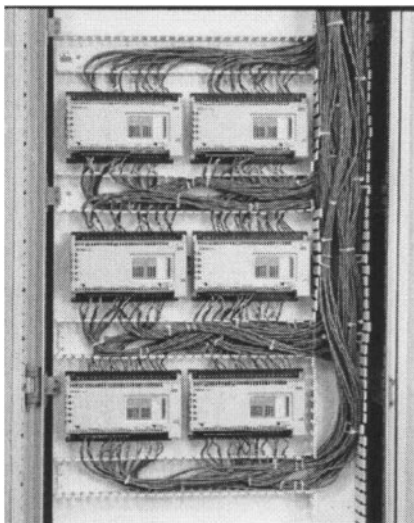
No. of Measurements	Channels per Card	No. of I/O Cards	Equiv. Fieldbus Cards
64	8	8	1

**Reduction in the Number of I/O Cards**

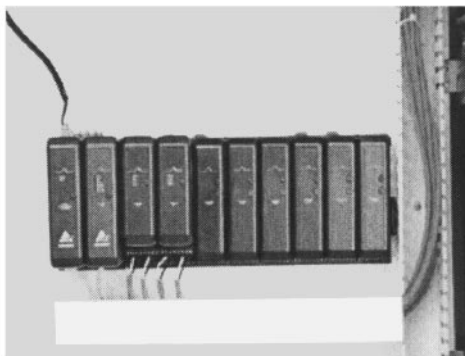
Traditional I/O cards consist of digital input, digital output, analog input, and analog output. Each card may, for this example, accommodate up to eight channels. With fieldbus, analog input, analog output, digital input, and digital output function blocks reside in the device, thus eliminating the need for the traditional I/O cards. A fieldbus card, which is used to interface with the fieldbus devices, may, for example, accommodate 64 function blocks from multiple fieldbus devices. To illustrate the impact fieldbus has in the number of I/O cards required, compare a traditional I/O layout of 64. Assuming that two function blocks are supported by each fieldbus transmitter, a single fieldbus connection equates to two analog input channels in a traditional installation. A traditional configuration of 64 analog inputs may require, in some cases, a total of eight I/O cards. The equivalent information accessed on fieldbus may require only one fieldbus interface card, as shown in the Table 7. This 87% reduction in the number of I/O cards will result in reduced system cost and space requirements for I/O cards.



**FIGURE 21** Example terminations for 16 fieldbus devices.



**Traditional I/O Footprint**



**Fieldbus controller and I/O Footprint**

**FIGURE 22** System footprint: comparison of traditional to fieldbus systems for the same number of field devices.

### Reduction in Instrument Room Space

The reduction of an instrument room for control is directly related to the I/O card reduction previously discussed. A visual comparison of a traditional installation to a fieldbus system with an equivalent I/O capability shows the dramatic reduction in instrument room space, which may be achieved using Foundation Fieldbus devices (Fig. 22).

## SUMMARY

---

Fieldbus technology clearly presents the opportunity for cost reductions in the following areas:

- installation labor and material costs of terminations
- installation labor and material costs of wiring
- reduction in the use of components such as I/O cards
- reduction in instrument control room panel space

Additional benefits that have tangible cost savings are in the following areas:

- reduction of start-up and commissioning costs
- integration of plantwide information to provide timely access to information
- automation of plant functions
- reduction of operating and maintenance costs (remote diagnostics, troubleshooting, spares)
- enable improvements in quality and safety
- maximize plant up time

It is too early to have hard numbers on the long-term benefit that the diagnostic and maintenance feature of fieldbus devices will provide these initial Foundation Fieldbus installations. However, the feeling of plants that have installed fieldbus is that these savings will match or exceed the installation savings that fieldbus has provided.

### Best Practices in Applying Fieldbus

1. Limit the number of fieldbus devices on a single fieldbus segment to 16. This will help ensure reasonable update rates of dynamic fieldbus parameters in the host interface.
2. Install on a single fieldbus segment no more than 12 fieldbus devices that are powered by the fieldbus segment. For an intrinsically safe installation, limit the maximum to four to six devices based on individual device power consumption.
3. Locate fieldbus devices on a single segment if control or calculations are distributed between these fieldbus devices.
4. Utilize a tree topology whenever possible for connecting fieldbus devices to a segment. Avoid daisy-chaining fieldbus devices since this will cause installation, commissioning, and maintenance problems.
5. Locate field junction boxes within 120 m of fieldbus devices that are to be on a fieldbus segment.
6. Install an approved barrier with the fieldbus power supply when fieldbus devices are to be installed in a classified area and the fieldbus segment is not protected by hard conduit or other approved materials.
7. Limit the fieldbus segment length to 1,900 m when using Type A cable (shielded twisted pair; 0.8 mm<sup>2</sup>).
8. Utilize a maximum of 16 function block input and output links between fieldbus devices on a single segment to provide reasonable loop execution rates and display update time at a host.
9. Purchase fieldbus devices that have been certified by the Fieldbus Foundation to be interoperable. Consider the function blocks and diagnostics supported by the fieldbus device when selecting a fieldbus device.
10. Select a control system that is specifically designed to support the interface, configuration, and diagnostics of fieldbus devices. In particular, the configuration system should utilize the DD provided by the device manufacturer to allow all fieldbus block parameters of a device to be accessed.
11. Include the location of power conditioners and terminators on fieldbus segment drawings and verify their installation before initially powering up the segment.
12. Continue shields throughout the fieldbus segment and have only one ground reference—at the source of power.
13. Estimate terminations and a wire reduction of 70–80% when 16 fieldbus devices are installed on each segment.
14. Plan on an 80–90% reduction in the number of I/O cards when 16 devices are included on a fieldbus segment. However, the cost of a fieldbus interface card may be higher than a traditional I/O card.
15. Take advantage of the fact that the rack room space for a complete fieldbus installation with 16 devices per segment will typically require only 30% of the space needed for a traditional installation.
16. Consider the fact that some projects have reported an 80–90% deduction in time for instrumentation drawings and instrument commissioning and checkout time over a traditional installation using fieldbus. However, on your first installation in a plant, plan on some of these savings being offset by additional time spent on fieldbus training.

## REFERENCES

---

1. Blevins, T., "Fieldbus Ushers In A New Era in Process Control," *Plant Services*, September 1998.
2. Blevins, T., J. Duffy, and R. Willems, "DCS Integration of Fieldbus," *ISA Conf. proceedings*, 1996.
3. Blevins, T., and W. Wojsznis, "Fieldbus Support for Process Analysis," *ISA Trans.* Vol. 35, pp. 177–183, 1996.
4. Instrument Society of America, "User Layer Technical Report for the Fieldbus Standard," ISA-TR50.02, Part 9-TR1, Tech. Rep. ISA/SP50-1993-389F, 1993.
5. Furness, H., "Distributed Control Functionality Moves Downstream," *Control Engineering*, December 1993.
6. International Electrotechnical Commission, Technical Committee No. 65, Industrial-Process Measurement and Control, Working Group 6, Committee Draft: Function Blocks for Industrial-Process Measurement and Control Systems, Part 1—General Requirements.
7. Fieldbus Foundation, Fieldbus Specification, Function Block Application Process—Part 1&2, FF-94-890, FF-94-8891.
8. Bialkowski, W. L., and A. D. Weldon, "The Digital Future of Process Control: Possibilities, Limitations, and Ramifications," *Tappi Journal*, Vol. 77, No. 10, October 1994.
9. Wheelis, J. D., and K. Zech, "Benefits Observed During Field Trials of an Interoperable Fieldbus," *ISA Conf. Proceedings*, 1994.

---

## BATCH CONTROL: APPLYING THE S88.01 STANDARD

by Thomas G. Fisher\*

### INTRODUCTION

---

The S88.01 standard [1] provides the basis for an object-oriented approach that fits very well with batch control and the automation of batch processes, including the development of objects that can be reused from project to project. Significant savings from applying these standards and guidelines have been demonstrated in many phases of batch control projects. Considerable emphasis is given in the standard to the separation of the recipe procedure (which tells how a batch should be made) from the equipment logic (which actually executes the batch). This separation is one of the major reasons that this standard has been so successful.

S88.01 is entitled Models and Terminology, but it is really a communications standard, although not in the sense of communications protocols over a local area network. S88.01 makes it possible for people to communicate about batch control by using a common language and a common set of models that describe batch control and batch manufacturing. S88.01, *Part 1—Models and Terminology*, was approved in February 1995. S88.01 became a joint ANSI (American National Standards Institute) and ISA (International Society for Measurement and Control) standard in October 1995. IEC 61512 [2] is a corresponding international standard from IEC (the International Electrotechnical Commission).

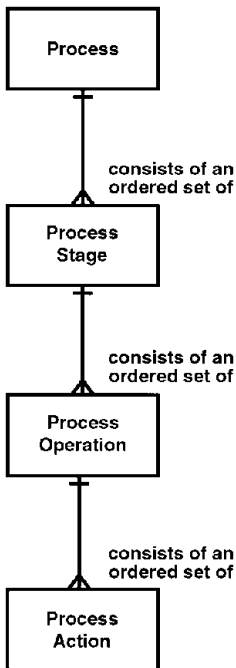
The NAMUR NE-33 guidelines [3] is often referenced in discussions about batch control. NAMUR is a group of German user companies, and it was founded in 1949 as an association of the measuring and control engineering departments of the chemical industry. Working Group 6, Batch Control, was founded in 1966. The first official attempt to set criteria and guidelines for batch automation and

---

\* Operations Technology Manager, Lubrizol Corporation, Wickliffe, Ohio.

related application software was made by this group under the chairmanship of R. J. Uhlig. In July 1985, they drew up a set of guidelines for batch control software that was entitled "Generation of Control Sequences for Batch Processes with Changing Recipes by Configuration, Using Predefined Functional Modules." This 1985 work formalized the approach of defining recipes by configuration, using predefined software building blocks (programs and data) that reflect and implement the fundamental operations (Grundfunktionen) of a batch process. These guidelines were published as a NAMUR status paper in 1987. The NE-33 guidelines, "Requirements to be Met by Systems for Recipe-Based Operations," were issued in May of 1992.

## DEFINITIONS



**FIGURE 1** Process model. (Copyright © ISA. Reprinted with permission. All rights reserved.)

A Batch Process is a process that leads to the production of finite quantities of material by subjecting quantities of input materials to an ordered set of processing activities over a finite period of time, using one or more pieces of equipment. The processing activities are defined in the various levels of the process model (see Fig. 1). The process model describes what is required by the batch process, and it is a way of organizing the subdivisions of a batch process in a hierarchical fashion.

The process consists of one or more process stages that are organized as an ordered set that can be serial, parallel, or both. A process stage usually results in a planned sequence of chemical or physical changes in the material being processed. An example of a process stage might be polymerize, that is, polymerize vinyl chloride monomer (VCM) into polyvinyl chloride. Each process stage consists of an ordered set of one or more process operations. A process operation usually results in a chemical or physical change in the material being processed. An example of a process operation for the polymerization process stage might be Charge (add demineralized water and add surfactants).

Each process operation can be subdivided into an ordered set of one or more process actions that carry out the processing required by the process operation. Process actions describe minor processing activities that are combined to make up a process operation. A typical process action for the Charge process operation might be Add (the required amount of demineralized water to the reactor).

A *Batch* has two meanings in S88.01. The first meaning of batch is the material that is being produced or that has been

produced by a single execution of a batch process. The second meaning of batch is an entity that represents the production of a material at any point in the process. Therefore, batch means both the material made by and during the process and also an entity that represents the production of that material. Batch is used as an abstract contraction of the words "the production of a batch."

*Batch control* refers to control activities and control functions that provide a means to process finite quantities of input materials by subjecting them to an ordered set of processing activities over a finite period of time, using one or more pieces of equipment.

A *recipe* is the necessary set of information that uniquely defines the production requirements for a specific product. The recipe tells the batch control system and/or the operator how to make product. A recipe usually exists for each intermediate and finished product that is to be produced.

*Equipment control* is the equipment-specific functionality that provides the actual control capability for an equipment entity, including procedural, basic, and coordination control, and that is not part of the recipe.

An *equipment entity* is a collection of physical processing and control equipment and equipment control grouped together to perform a certain control function or set of control functions.

## RECIPES

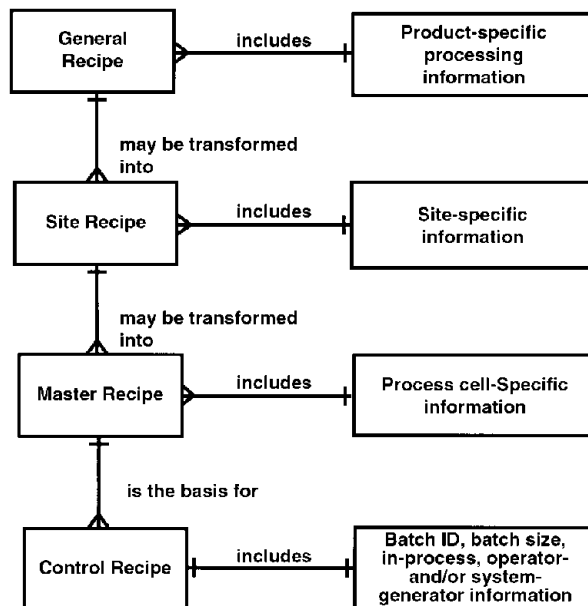
Recipes provide a way to describe products and how those products are to be produced. A product may be made in many different arrangements of equipment at many different sites. Recipes that are appropriate for one site or set of equipment may not be appropriate for another site or set of equipment. This can result in multiple recipes for a single product. Sufficient structure should be provided in the definition of recipes to allow tracing of the genealogy of any given recipe.

The recipe contains neither scheduling nor equipment control; it contains process-related information for a specific product. This concept allows batch processing equipment to make many different products without having to redefine equipment control for each product. This section discusses the four types of recipes that are covered in the S88.01 standard and the five categories of information that are contained in a recipe.

### Recipe Types

Fundamental to the practical application of recipes is the concept that different parts of an enterprise may require information about the manufacture of a product in varying degrees of specificity, because different recipients of the information use it for different purposes. Therefore, more than one type of a recipe is needed in an enterprise. The S88.01 standard defines only the general recipe, site recipe, master recipe, and control recipe (see Fig. 2). General and site recipes are substantially different than master and control recipes. The general and site recipes describe the technique (i.e., how to do it in principle). Master and control recipes describe the task (i.e., how to do it with actual resources).

Whether a particular recipe type actually exists, who generates it and where it is generated will vary from case to case and from enterprise to enterprise. An enterprise may choose not to implement one or more of the recipe types, and, depending on the specific requirements of an enterprise, other recipe types may exist.



**FIGURE 2** Recipe types. (Copyright © ISA. Reprinted with permission. All rights reserved.)

**General Recipes.** A general recipe is the parent of all lower-level recipes. A general recipe provides general manufacturing requirements in an equipment-independent format, and it applies to all sites within the enterprise.

**Site Recipes.** A site recipe is specific to a particular manufacturing site within the enterprise. A site recipe is derived from a general recipe, and little transformation is typically required. A site recipe is also in an equipment-independent format, and it is generally used for the following:

- planning
- costing
- long-term production scheduling

**Master Recipes.** A master recipe is derived from a site recipe, and a major transformation is expected to be required because the master recipe is specific to equipment at the manufacturing site. A master recipe may be targeted at a class of equipment or to specific pieces of equipment. Master recipes contain information that is needed for batch scheduling.

**Control Recipes.** A control recipe starts its life as a copy of a master recipe, and it is the recipe that is used to make a batch of product, whether automatically by a control system or manually by an operator. A control recipe is specific to a particular batch of product, and it may change as the batch progresses.

## Recipe Information Categories

Recipes contain the following categories of information: header, formula, procedure, equipment requirements and other information. The following discussion provides details regarding these categories.

**Header.** The administrative information in the recipe is referred to as the header. Typical header information may include the recipe and product identification, the version number, the originator, the issue date, approvals, status and other administrative information. For example, a site recipe may contain the name and version of the general recipe from which it was created.

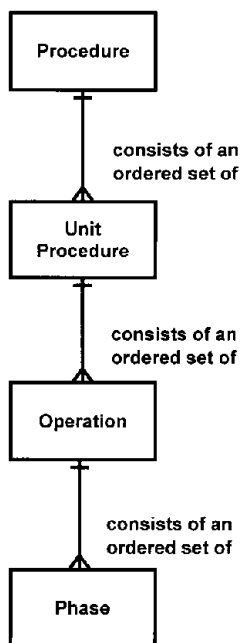
**Formula.** The formula is a category of recipe information that includes process inputs, process parameters, and process outputs.

A process input is the identification and quantity of a raw material or other resource that is required to make the product. In addition to raw materials that are consumed in the batch process in the manufacture of a product, process inputs may also include energy and other resources (e.g., manpower). Process inputs consist of both the name of the resource and the amount required to make a specific quantity of product. Quantities may be specified as absolute values or as equations based upon other formula parameters or the batch or equipment size. Process inputs may specify allowable substitutions, and they are expressed in the same basic form.

A process parameter details information such as temperature, pressure, or time that is pertinent to the product but that does not fall into the classification of input or output. Process parameters may be used as set points, comparison values, or in conditional logic.

A process output is the identification and quantity of a material and/or energy that is expected to result from one execution of the recipe. These data may detail environmental impact, and they may also contain other information (e.g., specification of the intended outputs in terms of quantity, labeling, and yield).

The types of formula data are distinguished to provide information to different parts of an enterprise, and they have to be available without the clutter of processing details. For example, the list of process inputs may be presented as a condensed list of ingredients for the recipe or as a set of individual ingredients for each appropriate procedural element in a recipe.



**FIGURE 3** Procedural control model.  
(Copyright © ISA. Reprinted with permission. All rights reserved.)

**Procedure.** The procedure defines the strategy for carrying out a process. The general and site recipe procedures are structured by using the levels described in the process model because these levels allow the process to be described in equipment-independent terms. The master and control recipe procedures are structured by using the procedural elements of the procedural control model (see Fig. 3) because these procedural elements have a relationship to equipment.

The recipe creator is limited to the use of procedural elements that have been, or will be, configured and made available for use in creating a procedure. He or she may use any combination of these procedural elements to define a procedure. Determination of which of these procedural elements that may be part of the procedure is an application specific design decision that is based on many factors, including the capabilities of the controls and the degrees of freedom appropriate for the recipe creator in a given application.

The procedural control model shown in Fig. 3 defines the control that enables equipment to provide the process functionality required by the batch process. Procedural control is made up of procedural elements, and they are combined in a hierarchical manner in order to accomplish the task of a complete process as defined by the process model.

A procedure is the highest level in the hierarchy, and it defines the strategy for carrying out a major processing action (e.g., making a batch). A procedure is defined in terms of an ordered set of unit procedures. An example of a procedure is “Make PVC.”

A unit procedure consists of an ordered set of operations that cause a contiguous production sequence to take place within a unit. An example of a unit procedure in the Make PVC procedure might be “Polymerize VCM.”

An operation is an ordered set of phases that define a major processing sequence that takes the material being processed from one state to another, and

a chemical or physical change is usually involved. An example of an operation in the Polymerize VCM unit procedure might be React (Add VCM and catalyst, Heat, and Wait for the reactor pressure to drop).

A phase is the smallest element of procedural control that can accomplish a process oriented task. The intent of the phase is to cause or define a process oriented action, while the logic or set of steps that make up a phase is equipment specific. An example of a phase in the React operation might be Add (Add catalyst).

**Equipment Requirements.** Equipment requirements constrain the choice of the equipment that will eventually be used to implement a specific part of the procedure. In the general and site recipes, the equipment requirements are typically described in general terms, such as allowable materials and required processing characteristics. The guidance from and the constraints imposed by equipment requirements will allow the general or site recipe to eventually be used to create a master recipe that targets appropriate equipment. At the master recipe level, the equipment requirements may be expressed in any manner that specifies allowable equipment in process cells. If trains have been defined, then it is possible for the master recipe (and the resulting control recipe) to be based on the equipment of the train rather than the full range of equipment in the process cell. At the control recipe level, the equipment requirements are the same as, or a subset of, the allowable equipment in the master recipe. The control recipe may be used to include specific allocations of process cell equipment, such as Reactor R-501, when this becomes known.

**Other Information.** The data that are contained in the Other Information category are usually one of the following:

- recipe-dependent safety comments, but not MSDS (material safety data sheet)
- recipe-dependent compliance comments
- data-collection requirements
- special reporting requirements

## EQUIPMENT ENTITIES

---

This section discusses equipment entities that are formed from the combination of equipment control and physical equipment. This combination results in four equipment entities: process cells, units, equipment modules, and control modules. Guidelines for structuring these equipment entities are also discussed.

When the terms process cell, unit, equipment module, and control module are used, they generally refer to the equipment and its associated equipment control. Whether equipment control in an equipment entity is implemented manually or by way of automation, it is only through the exercise of equipment control that the equipment can produce a batch.

The notion of equipment control being part of an equipment entity is to be understood logically because it is not a statement of the physical implementation of equipment control. However, equipment control for a particular equipment entity must be identifiable.

This interaction of equipment control and physical equipment is described purposely without any reference to language or implementation. The intent is to describe a framework within which equipment control and physical equipment may be defined and discussed.

### Equipment Control

Three types of control are defined: basic control, procedural control, and coordination control.

**Basic Control.** Basic control is dedicated to establishing and maintaining a specific state of equipment and process, and it includes the following:

- regulatory control
- interlocking
- monitoring
- exception handling
- repetitive discrete or sequential control

**Procedural Control.** Procedural control is a characteristic of batch processes, and it is used to direct equipment-oriented actions. Procedural control is based on the procedural control model.

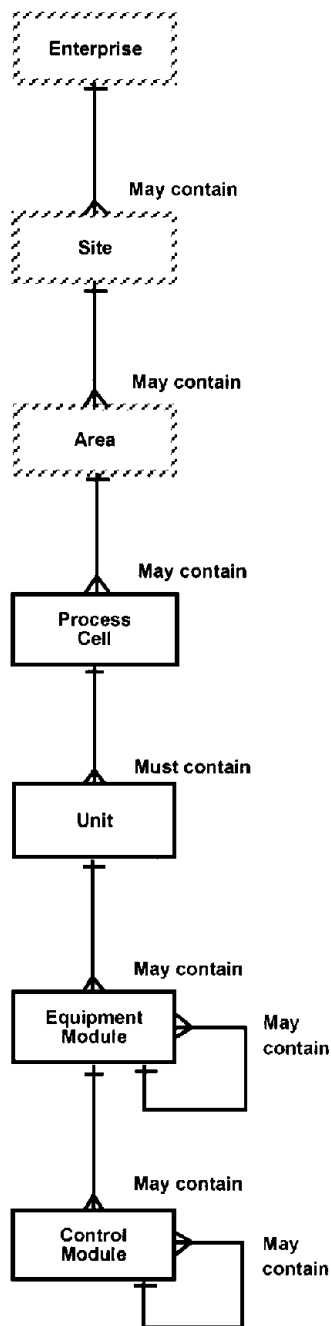
**Coordination Control.** Coordination control directs, initiates, and/or modifies the execution of procedural control and the utilization of equipment entities. The following are some examples of coordination control:

- supervising the availability of equipment
- allocating equipment to batches
- arbitrating requests for allocation
- coordinating common resource equipment

More coordination control is to be expected at the process cell level, while little coordination control is typically found at the control module level.

### Physical Model

The physical model (see (Fig. 4) describes what equipment is available for the batch process. The physical assets of an enterprise that is involved in batch manufacturing are usually organized in a hierarchical fashion as described by the physical model. Lower-level groupings are combined to form



**FIGURE 4** Physical model. (Copyright © ISA. Reprinted with permission. All rights reserved.)

higher levels in the hierarchy. In some cases, a grouping within one level may be incorporated into another grouping at that level.

The model has seven levels, starting at the top with an enterprise, a site and an area. These three levels are frequently defined by business considerations, and they are not modeled further in S88.01. The three higher levels are part of the model to properly identify the relationship of the lower-level equipment to the manufacturing enterprise.

The lower four levels of this model refer to specific equipment types. An equipment type in Fig. 4 is a collection of physical processing and control equipment grouped together for a specific purpose. This grouping is usually done to simplify operation of the lower-level equipment by treating it as a single larger piece of equipment.

Equipment entities are defined for the lowest four levels of the physical model (i.e., process cell, unit, equipment module, and control module).

**Control Module.** A control module is a collection of sensors, actuators, other control modules, and associated processing equipment. A control module acts as a single entity from a control standpoint, and it is the direct connection to the process through its sensors and actuators. A control module cannot execute procedural elements. The following are some examples of control modules:

1. A flow control loop that operates by means of the set point of the controller.
2. An on-off automatic block valve with limit switches that operates by means of the set point (e.g., open and close) of the valve.
3. A header that contains several automatic block valves and that directs flow to different destinations based on a set point to the header.

**Equipment Module.** An equipment module is a collection of control modules and/or other equipment modules. An equipment module can carry out a finite number of minor processing activities (i.e., phases), and it contains all the necessary processing equipment that is needed to carry out these processing activities. The following are some examples of equipment modules:

1. A weigh tank that is shared by multiple units but that can only be used by one unit at a time.
2. A filter that is a permanent part of a particular unit.
3. An ingredient supply system that is shared by multiple units and that can be used simultaneously by all units.

Control modules and equipment modules are used because these combinations of various instrument functions can be addressed as a single entity. The alternative to these equipment entities is to deal with the individual equipment-oriented checks and actions.

**Unit.** A unit is usually centered on a major piece of process equipment, and it frequently operates on or contains the complete batch. Although a unit may operate on or contain only a portion of a batch, it cannot operate on or contain more than one batch at a time.

Units are the primary object for automatic control, and they have a direct relationship with unit procedures and operations. A unit is made up of control modules and/or equipment modules, and there will often be multiple units involved in making a batch. Control modules and equipment modules can exist as:

- permanently included parts of a unit
- temporarily attached parts of a unit
- totally separate from a unit

When control modules and/or equipment modules are not part of a unit, they may be connected to a unit, and then they may be commanded like any other object in the unit.

**Process Cell.** A process cell is a logical grouping of equipment that is required for the production of one or more batches. A process cell may contain more than one grouping of equipment that is needed to make a batch. That grouping is referred to as a Train. The equipment that was actually used to execute a batch is referred to as the Path. A process cell frequently contains more than one batch at a time.

## Partitioning Equipment Entities

This section discusses the general principles involved in partitioning a process cell into equipment entities that can carry out specified processing activities or equipment-specific actions [4], [5].

The physical process cell design can greatly influence the implementation of batch control. Minor differences in the physical system can dramatically affect the organization of equipment entities and procedural elements.

All control-related sections of the standard assume that the process cell in question (both physical equipment and related control activities) has been subdivided into well-defined equipment entities such as units, equipment modules, and control modules. Effective subdivision of the process cell into well-defined equipment entities is a complex activity, and it is highly dependent on the individual requirements of the specific environment in which the batch process exists. Inconsistent or inappropriate equipment subdivisions can compromise the effectiveness of the modular approach to recipes suggested by this standard, and they may result in a solution that:

- is difficult to support and enhance
- fails to exploit the inherent flexibility of the plant
- requires control system experts to assist process personnel when they are developing new recipes or modifying existing recipes

Subdivision of the process cell requires a clear understanding of the purpose of the process cell's equipment. Such understanding allows the identification of equipment entities that can work together to serve an identifiable processing purpose.

**Process Cells.** The subdivision of a process cell usually follows the principles listed below:

1. The function any equipment entity serves in product processing must be clear and unambiguous.
2. The function performed by the equipment entity must be consistent in terms of processing task, and it should be usable for that task no matter what product is being manufactured at a given time.
3. Subordinate equipment entities should be able to execute their task(s) independently and asynchronously because this allows the highest level equipment entity to orchestrate the activities of its subordinates.

4. Interactions between equipment entities should be minimized. While planned interaction is periodically necessary, each equipment entity should perform its functions while influencing the functioning of other equipment entities as little as possible.
5. Equipment entities must have clear boundaries.
6. A consistent basis is required for the definition of equipment entities. An operator subsequently interacting with similar equipment entities should be able to do so naturally and without confusion.
7. Necessary interaction between equipment entities is, insofar as possible, coordinated by equipment entities at the same level or at the next higher level.
8. The process cell may have multiple units and/or equipment modules of the same type that operate in parallel.
9. The number of units in the process cell will not necessarily be the same as the number of vessels in the process cell.
10. The equipment entities that are defined should perform essentially independent tasks.

**Units.** The definition of a unit requires knowledge of the major processing activities, as well as the equipment capabilities. The following guidelines apply:

1. One or more major processing activities (i.e., operations), such as reaction or crystallization, may take place in a unit, but only one may be active at a time.
2. Units should be defined such that they operate relatively independently of each other.
3. A unit can operate on only one batch at a time.
4. A unit can operate on all or a part of a batch.
5. A unit may contain equipment modules and control modules, but a unit may not contain another unit.

**Equipment Modules.** The definition of an equipment module requires knowledge of specific minor processing activities and equipment capabilities. The following guidelines apply:

1. An equipment module can carry out a finite number of minor processing activities, such as dosing and weighing.
2. These activities are typically centered around a set of process equipment.
3. Collections of control modules can be defined as equipment modules or as control modules. If the collection executes one or more equipment phases, then it is an equipment module.
4. An equipment module may be part of a process cell, a unit, or another equipment module.

**Control Modules.** A control module is the lowest level of grouping that operates as a single entity. The following guidelines apply:

1. A control module executes basic control.
2. A control module cannot execute procedural control.

### Procedural Control Model/Physical Model/Process Model Relationship

The general relationship between the procedural control model, the physical model, and the process model is illustrated in Fig. 5. This mapping of procedural control with individual equipment provides the processing functionality described in the process model.

The concept of equipment capabilities and usage of these capabilities to accomplish processing tasks is a major point of the S88.01 standard. The procedural control capability of equipment entities is the mechanism that enables this. The procedural control may be entirely defined as part of the equipment entity—or it may be based on procedural information passed on to the equipment entity from the recipe procedure.

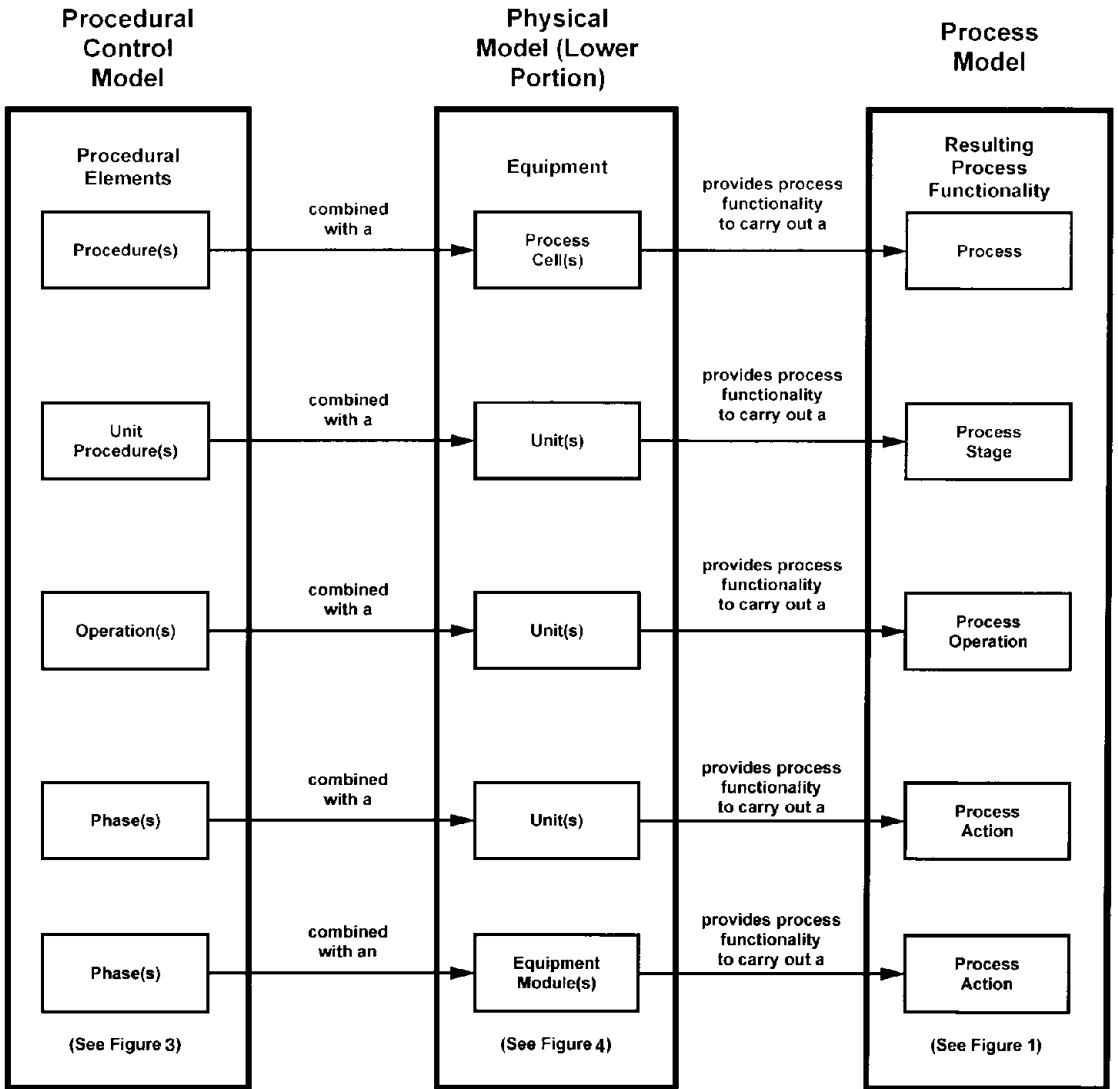
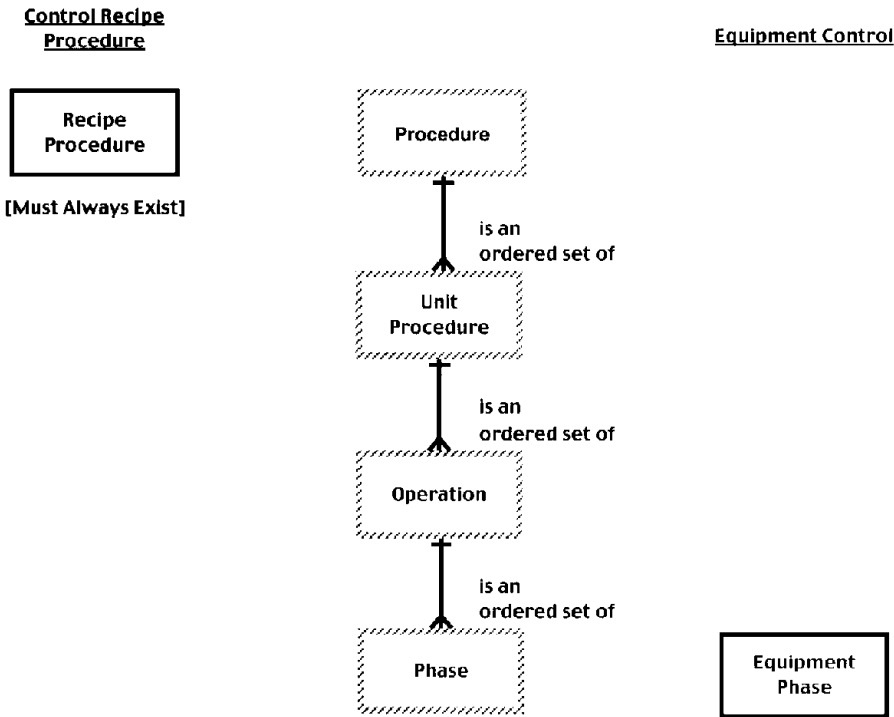


FIGURE 5 Procedural control model/physical model/process model relationship. (Copyright © ISA. Reprinted with permission. All rights reserved.)

### RECIPE PROCEDURE/EQUIPMENT CONTROL SEPARATION

This concept of separating the recipe that describes how the batch is to be made from the equipment that is actually used to make the batch results in the following advantages:

1. Improves recipe transportability because the recipe does not contain all the equipment-specific logic.



**FIGURE 6** Recipe procedure/equipment control separation. (Copyright © ISA. Reprinted with permission. All rights reserved.)

2. Makes the recipe more flexible because it is applicable to a wider range of batch control equipment, even if the batch control equipment operates differently. As an example, a recipe phase that adds an ingredient to a batch doesn't care if the fluid is metered with a flow meter or a weigh tank.
3. Simplifies recipe validation because the equipment logic does not have to be revalidated every time a change is made to a recipe.
4. Makes software modularity feasible. The goals of software modularity can be accomplished by:
  - Making the creation or modification of a recipe easy enough so that control systems experts are not needed to develop and maintain recipes.
  - Using predefined and pretested building blocks (i.e., standard software modules) for implementing equipment control.
  - Making it easy to link the recipe with these building blocks.

Figure 6 shows the separation between the control recipe procedure and equipment control. The control recipe procedure must contain at least one procedural element, which is the recipe procedure. Equipment control must also contain at least one procedural element that provides the linkage needed to operate the physical equipment. For the example described in Fig. 6, this procedural element is assumed to be the equipment phase.

The control recipe procedure might not include recipe unit procedures, recipe operations, and recipe phases. Such a recipe procedure must then be linked (by reference) to an equipment procedure in equipment control if batches are to be executed. Whenever a procedural element (i.e., recipe

procedure, recipe unit procedure, recipe operation, or recipe phase) is linked to equipment control, it must exist as that recipe procedural element (such as a recipe operation) and as that equipment procedural element (such as an equipment operation). Whenever recipe phases are used in the control recipe procedure, recipe phases are linked to equipment phases.

When recipe unit procedures, recipe operations, and recipe phases are not used as part of the control recipe procedure, the use of lower-level equipment procedural elements (some or all) as part of equipment control can provide a modular structure to the equipment control.

### Control Recipe Procedure/Equipment Control Linking

There must be some method to link the control recipe procedural elements with the equipment procedural elements. This linking is done by associating the recipe procedural elements with equipment procedural elements. In this way, the call for a certain processing function is separated from equipment control, and this enables the same recipe procedural element to use different equipment procedural elements, depending on what equipment the recipe addresses.

An equipment phase may be initiated by things other than the execution of a control recipe (e.g., by the request of another unit or on the request of an operator). The independent execution of a phase may be useful for handling exception conditions, during start-up or maintenance and/or to prepare a unit for production.

If unit procedures, operations, and phases are part of the control recipe procedure, linking (by reference) of the control recipe procedure to equipment control is done at the phase level (see Fig. 7). This drawing applies to one control recipe.

In a batch control system, this linking is accomplished with a phase logic interface. A phase logic interface is a layer of software that provides the interface between the recipe procedure and the corresponding equipment procedural element.

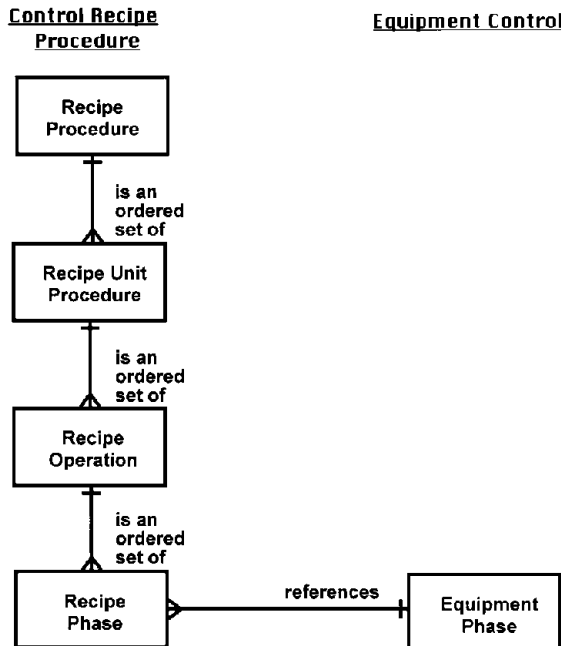


FIGURE 7 Recipe procedure/equipment control linking example. (Copyright © ISA. Reprinted with permission. All rights reserved.)

## Control Recipe/Equipment Procedural Elements

The following are typically associated with recipe procedural elements:

- a description of the functionality required
- formula and other parameter information specific to the procedural element
- equipment requirements specific to the procedural element

The equipment procedural element to be linked typically has the following:

- an identification that can be referenced by the recipe procedural element or a higher-level equipment procedural element
- a description of the functionality that is provided
- variables that can receive the formula and other parameter information from the recipe
- execution logic

In order for a recipe procedural element to be able to reference an equipment procedural element, it must have an identification that enables the element to be correctly linked. In other cases, it must reference or include other recipe procedural elements and a specification of the execution order of those procedural elements.

---

## PROCESS AND CONTROL ENGINEERING

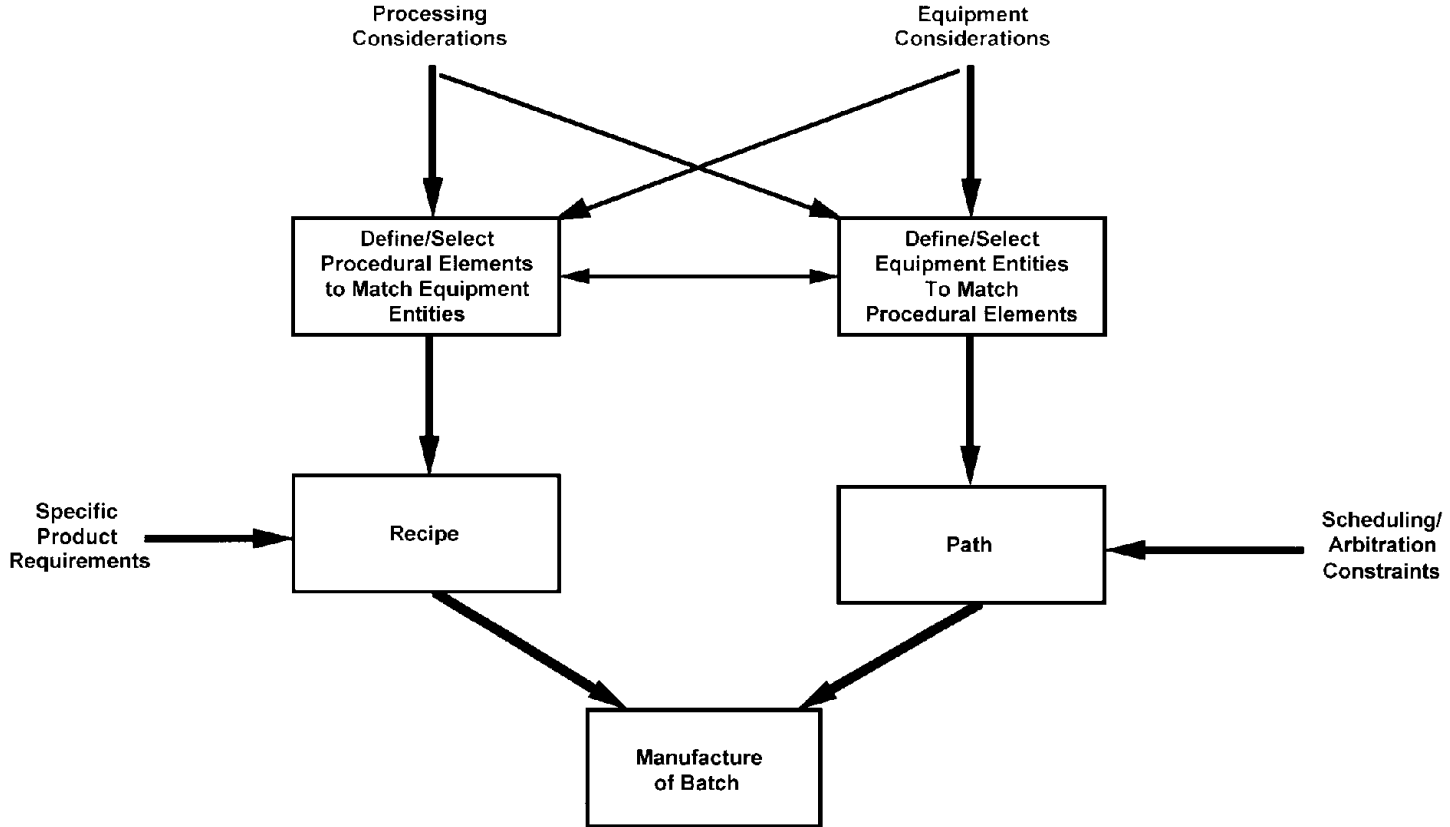
In order for required processing functions to be properly carried out in a batch manufacturing environment, the equipment structure needed, the process functionality, and the exception handling for that equipment have to be fully developed. This requires a coordinated engineering effort that continues from initial definition through the life of the batch processing facility. This section describes the process and control engineering needed for the design of the controls needed to support the recipe hierarchy, the definition of equipment capability, and the development of the functionality required in the procedures to produce a batch.

Process and control engineering is needed at the general and site recipe levels to describe procedures, process stages, process operations, and process actions and at the master recipe level to describe recipe procedures, recipe unit procedures, recipe operations, and recipe phases.

The precise definition of appropriate procedural elements and equipment entities is an iterative process. The dual work process is illustrated in Fig. 8. Considerations affecting one decision process also affect the other. Processing considerations are the primary input to the definition (or selection) of procedural elements that will characterize functionality for associated equipment entities. Since the functionality defined will be affected by the equipment used, equipment considerations must be a secondary input. In the same way, equipment considerations form the primary input and processing considerations form the secondary input when making the definition (or selection) of equipment entities.

Recipes can be constructed by using these procedural elements and specific product information. The equipment entities are arranged into a path that is determined by scheduling and taking into account arbitration constraints. The combination of the results of these activities provides a framework within which a batch of material can be manufactured.

Process and control engineering also includes the development and revision of the equipment phases corresponding to the recipe phases that are used to define the recipe. As far as possible, recipe and equipment phases should be defined such that any reasonable functionality of a unit can be expressed in terms of these phases. They should generally not be tailored to a set of known recipes. Then, new recipes can in most cases be written by using existing recipe phases that reference existing equipment phases. The development and revision of recipe and equipment phases is an ongoing activity that provides ongoing support to the batch manufacturing facilities. This activity is the result of the ongoing drive for continuous improvement and the periodic addition of new process technology.



**FIGURE 8** Definition of procedural elements and equipment entities. (Copyright © ISA. Reprinted with permission. All rights reserved.)

## CONTROL SYSTEM FUNCTIONAL SPECIFICATIONS

The models and terminology that are defined in S88.01 can be used as the basis for developing vendor-neutral or vendor-specific functional specifications for batch control. By the application of object-oriented techniques, reusable modules can be developed that can then be used from project to project.

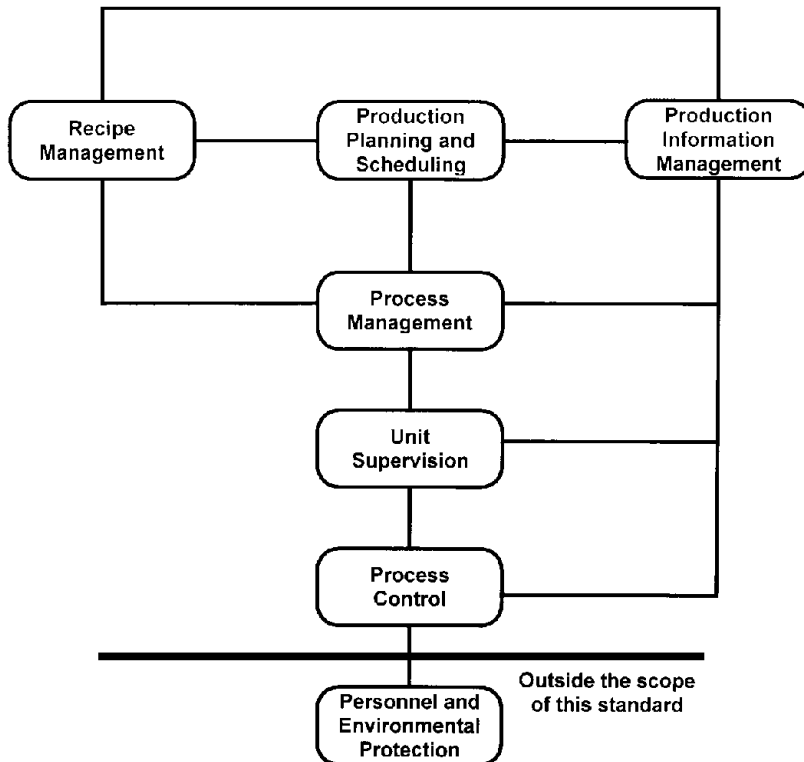
### What Is Needed To Define Batch Control

The following is the information that is needed in order to define the batch control requirements for a particular project:

- how the product is to be made (i.e., recipes)
- what equipment is to be controlled (i.e., equipment entities)
- what control is needed (i.e., control activities and control functions)

Recipes and equipment entities were discussed in previous sections. Control activities and control functions are defined by the control activity model (see Fig. 9). This model provides the basis for defining the control functionality that is required.

The control activity model is the cornerstone of the S88.01 standard. These control activities are the ones that are necessary to manage and control production in a batch manufacturing plant. The



**FIGURE 9** Control activity model. (Copyright © ISA. Reprinted with permission. All rights reserved.)

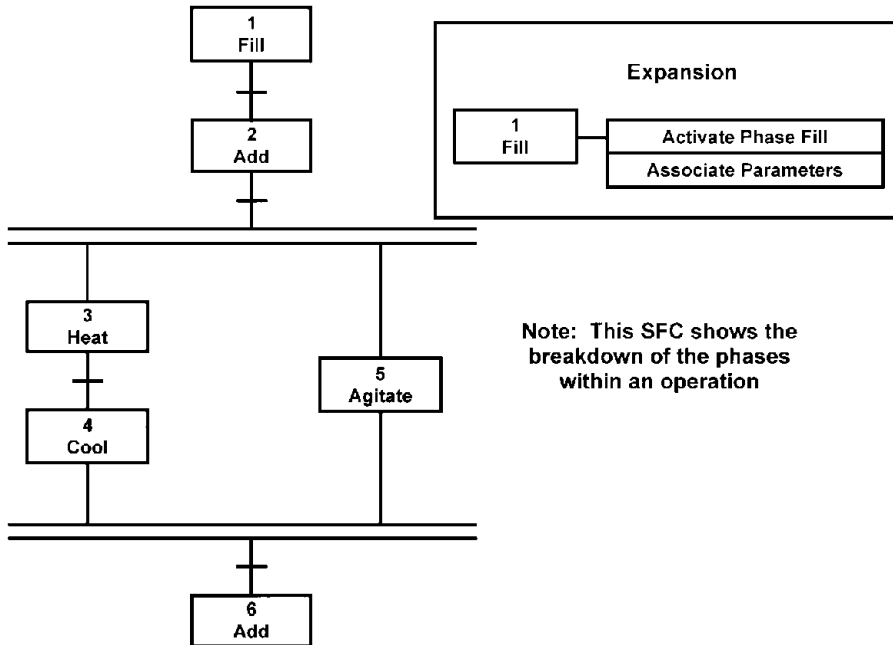


FIGURE 10 Using sequential function charts (SFCs) to depict recipe procedures.

control activity model is used to define what has to be accomplished to do batch control, whether the batch control is done automatically or manually. The control activities shown relate to real needs in a batch manufacturing environment. For example, the need to have control functions (note that control activities are broken down into control functions) that can manage general recipes, manage site recipes, and manage master recipes implies a need for the Recipe Management control activity.

**Recipe Management.** For control system functional specifications, the primary recipe of interest is the master recipe. The master recipes that are needed for the project can be developed by using the techniques that are defined in S88.01. The user must decide on the language that is used for recipe development and user presentation. An example using sequential function charts (SFCs) is shown in Fig. 10. This figure shows the breakdown of the phases in a React operation.

**Production Planning and Scheduling.** The user must decide how batch scheduling will be handled (i.e., manual or automatic). The format of the batch schedule must be determined in order to minimize any mismatch between the control function that supplies the batch schedule and the control system that uses the batch schedule. In addition, the functional specification should define what information must be fed back to the scheduling control function.

**Production Information Management.** A list of the information that must be collected from the control system must be developed, along with the format of that information. In addition, the functional specification should define what summary information must be supplied to higher-level business processes.

**Equipment Entities.** The three higher-level control activities that were described above communicate directly with the Process Management control activity. Process Management and the lower-level control activities are typically dealt with as part of Equipment Control.

Process Management maps to the process cell equipment entity. The process cell and the equipment that is contained within the process cell must be partitioned into equipment entities that correspond to the three lowest levels of the physical model:

- unit
- equipment module
- control module (includes safety interlocks)

The relationships between these equipment entities must also be specified.

The behavior of the equipment entities that were defined above must be specified so that the control functionality that was defined using the Control Activity Model can, in fact, be accomplished. This behavior may be defined by using a combination of the following:

- basic control
- procedural control
- coordination control
- data collection
- operator interface requirements (not discussed here)

## Equipment Entity Details

The following are things that must be dealt with as part of the definition of the equipment entities:

- phase logic
- allocation and arbitration
- unit-to-unit synchronization
- modes and states
- data collection
- exception handling

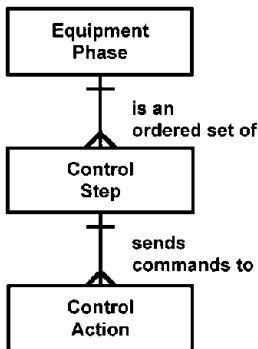


FIGURE 11 Equipment phase subdivisions.

**Phase Logic.** Equipment phases are usually broken down into control steps and control actions. The steps and transitions that are described in IEC 60848 or IEC 61131-3 document one method for defining the subdivisions of a phase (see Fig. 11).

The following are some things that control steps can do:

- enable and disable regulating and state-oriented types of control actions
- specify the set points and initial output values of control actions
- set, clear, and change alarms and other limits
- set and change controller constants, controller modes, and types of algorithms
- read process variables (e.g., alarm limits, set points, and controller status)
- send a message to an operator

Some of the categories of control actions are:

- control actions that perform an actuating element algorithm

- control actions that perform a regulating type of control algorithm or a state-oriented type of control algorithm
- control actions that perform arithmetic calculations
- control actions that communicate with people and/or other equipment
- control actions that make decisions and then control the direction and timing of higher-level control components

**Allocation and Arbitration.** Allocation is a form of coordination control that assigns a resource to a batch or unit. Allocation determines the routing or the “path” of the batch through the units. This patch may be dynamic (i.e., the path may change from batch to batch) depending on unit availability at the time the batch is executed.

Allocation must deal with common resources. When more than one unit can acquire or request the services of a single resource, that resource is a common resource. Common resources are often present with complex batch processes, and they are often implemented as equipment modules or control modules.

Common resources may be either exclusive-use resources or shared-use resources. When a resource is designated as an exclusive-use resource, only one unit can use that resource at a time (i.e., it is exclusive to that unit). An example is a shared weigh tank that can be used by many reactor units but that can only be used to charge one reactor at a time. This common resource must be allocated properly to avoid unit downtime while the unit is waiting for the use of the resource.

When a resource is designated as a shared-use resource, several units may use the resource at the same time. An example is a raw-material distribution system that is capable of delivering material to more than one unit at a time. If the capacity of the shared-use resource is limited (e.g., a maximum flow capability of a steam header), controls must be put in place to ensure that the resource’s capacity is not exceeded. Controls must also be put in place to ensure that one unit does not improperly shut off or deactivate a shared-use resource while other units are using the resource.

Arbitration is a form of coordination control that determines how a resource should be allocated when there are more requests for the resource than can be accommodated at one time. Arbitration is required when there are multiple requestors for a resource. Then some method must be put in place to resolve contention for the resource according to some predetermined algorithm and to provide definitive routing or allocation direction.

**Unit-To-Unit Synchronization.** The need for unit-to-unit synchronization is very common when transfers are made between units. This synchronization is usually handled with a Transfer operation in one unit and a Receive operation in the other unit. However, the actual communication between the units occurs at the equipment phase level. At least one commercial system handles transfers as part of the routing or path.

Figure 12 shows the communication between the Transfer operation in an esterification unit and the Initialize and Receive operations in a stripping unit.

**Modes and States.** A mode is defined as:

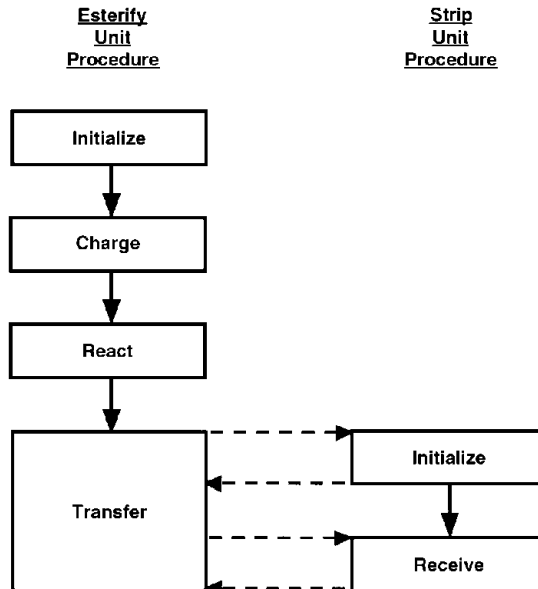
- the manner in which the transitions of sequential functions are carried out within a procedural element or
- the accessibility for manipulating the states of equipment entities manually or by other types of control

The modes that are defined in S88.01 are shown in Table 1.

Users must deal with mode propagation. For an example, a unit procedure changes to the semi-automatic mode. Should all lower-level procedural elements in that unit go to the semi-automatic mode? Propagation may be from higher to lower and vice versa.

**TABLE 1** Modes Defined in S88.01

Mode	Behavior	Command
Automatic (procedural control)	The transitions within a procedure are carried out without interruption as appropriate conditions are met.	Operators may pause the progression, but they may not force transitions.
Automatic (basic control)	Equipment entities are manipulated by their control algorithm.	The equipment cannot be manipulated directly by the operator.
Semi-automatic (procedural control only)	Transitions within a procedure are carried out on manual commands as appropriate conditions are fulfilled.	Operators may pause the progression or redirect the execution to an appropriate point. Transitions may not be forced.
Manual (procedural control)	The procedural elements within a procedure are executed in the order specified by an operator.	Operators may pause the progression or force transitions.
Manual (basic control)	Equipment entities are not manipulated by their control algorithm.	Equipment entities may be manipulated directly by the operator.

**FIGURE 12** Unit-to-unit synchronization example.

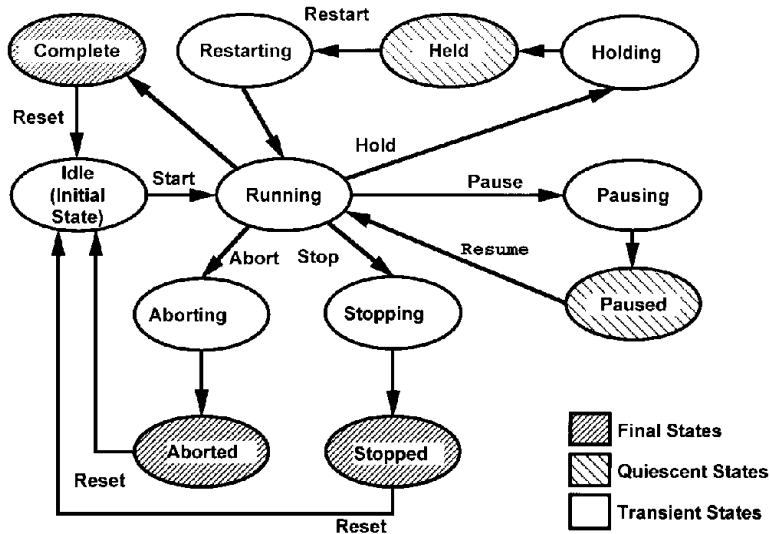


FIGURE 13 State transition diagram. (Adapted from S88.01, ISA.)

State is the condition of an equipment entity or of a procedural element at a given time. Some examples of states for equipment entities are On, Off, Closed, Open, Failed Tripped and Available. Some examples of states for procedural elements are Running, Holding, Paused, Stopped, Aborted, and Complete.

Procedural elements also have commands that move the procedural elements from one state to the next. Some examples of the commands that are applicable to procedural elements are Start, Hold, Pause, Stop and Abort. An example of the relationship between the procedural element states and the procedural element commands is shown in Fig. 13.

Users must also deal with state propagation. For example, a unit procedure moves to the Held state. Should all procedural elements in that unit go to the Held state? Should all procedural elements in *all units* go to the Held state? Propagation may be from higher to lower and vice versa.

**Data Collection.** Data collection is present in the lower four elements of the control activity model:

- process management: collect batch and process cell information
- unit supervision: collect batch and unit information
- process control: collect data
- personnel and environmental protection: not specified by S88.01, but collect data

A mechanism must exist to specify what data must be collected. These data must be related to the batch, to the equipment entity, and to the procedural element. Data must be collected on those things that are expected to happen during the course of the batch, but they must also be collected on those things that are not expected to happen. For example, if the recipe procedure is changed during the execution of a batch, data on any new recipe procedural elements and their execution must be collected.

**Exception Handling.** An exception is an event that occurs outside the normal or desired behavior. Exceptions can occur at all levels in the control activity model, and they may be part of procedural control, basic control, and coordination control. Exception handling typically accounts for a very large portion of the control definition (i.e., 50–80%).

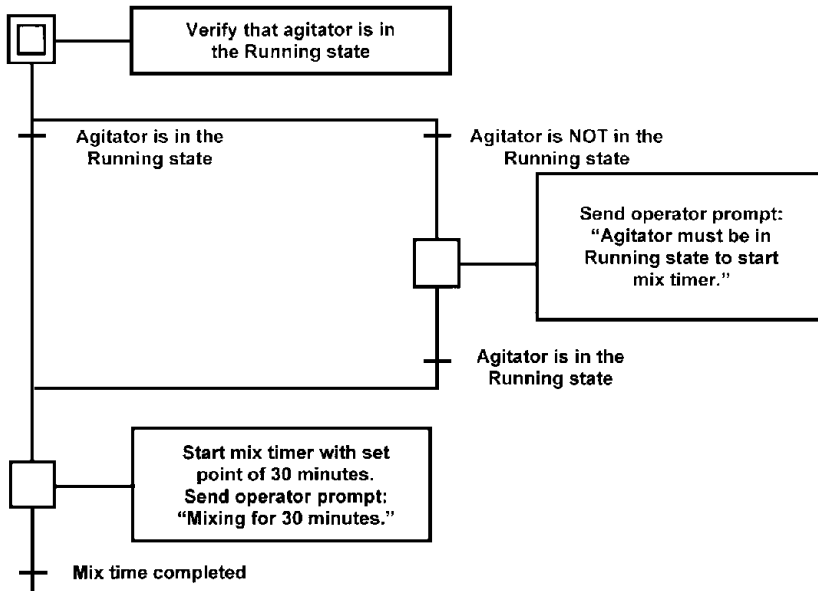


FIGURE 14 Exception handling in phase logic.

An event must be detected, then evaluated, and a response generated. Exceptions may affect the modes and states of equipment entities and procedural elements. For example, high pressure in a reactor could lead to the exception handling function transferring the process to the Stopped state.

The following are some different ways of implementing exception handling:

1. Implement exception handling that changes infrequently in equipment control.
2. Implement exception handling that changes frequently in the recipe.
3. Consider implementing product-related exceptions in equipment control using formula parameters.

If exception handling is implemented in the recipe, it should be as simple as possible and implemented at the recipe phase level, if possible.

Most exception handling is implemented in equipment control. The following are some different ways of implementing exception handling in equipment control:

1. Implement directly in phase logic.
2. Implement in phase logic using states.
3. Implement in control actions.

Figure 14 shows an example in which exception handling is implemented directly in phase logic. This method is effective if the exception handling functions are relatively simple, but it can get very complex if the exception handling functions are complex.

Exception handling is usually best done with states. The user must define what states are needed, but the states that are defined in S88.01 will meet most needs. The user must also define the conditions that cause state transitions (i.e., commands). If the states that are defined in S88.01 are used, then the equipment phase should contain the following logic:

- running logic
- holding logic

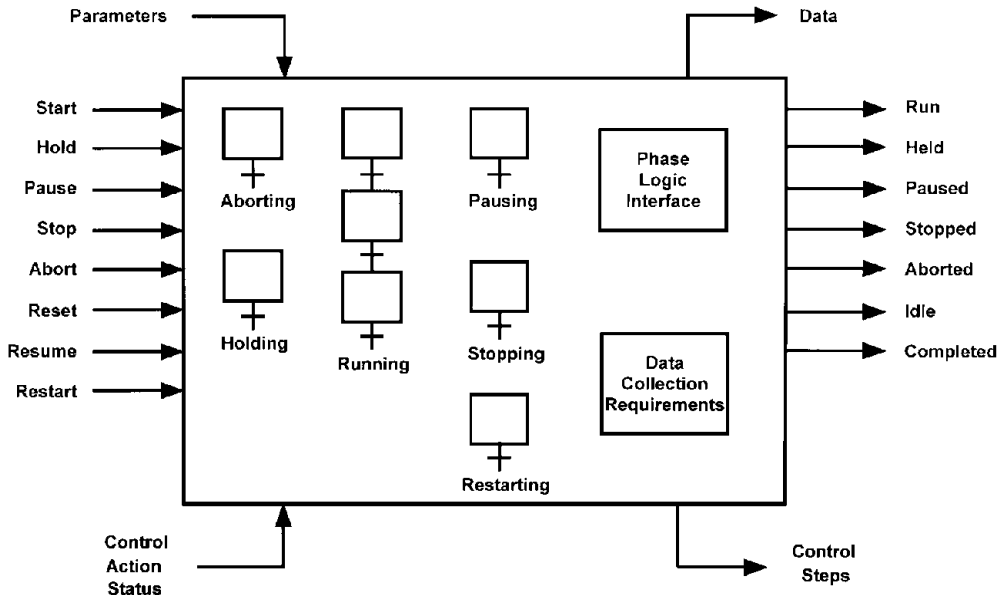


FIGURE 15 Equipment phase object.

- pausing logic
- aborting logic
- stopping logic
- restarting logic

An example of an equipment phase is shown in Fig. 15. Commands come in the left side of the phase block. The current state of the phase is shown in the status signals that come out of the right side of the block. Parameters come from the recipe, and data are fed back to higher-level elements. Commands (control steps) to control actions come out of the phase block on the bottom right side. Feedback on the status of the control actions comes back into the phase block on the bottom left side.

The phase logic interface is shown within the equipment phase. The phase logic interface is responsible for interfacing to the recipe and for enforcing the states of the phase. Data-collection requirements for the equipment phase are also shown. The logic for the various states is included within the equipment phase.

Figure 16 shows an example of exception handling in control actions. A safety interlock and a process interlock are both examples of control actions. When either of these interlocks trip, they can affect the operation of other control actions and phase logic.

## SUMMARY

S88.01 is a good tool for improving batch control-related communications between users and vendors and for developing functional specifications. This standard also provides the basis for actual implementation of the batch control system. S88.01 is a models and terminology standard, so many of the implementation details are left up to the user.

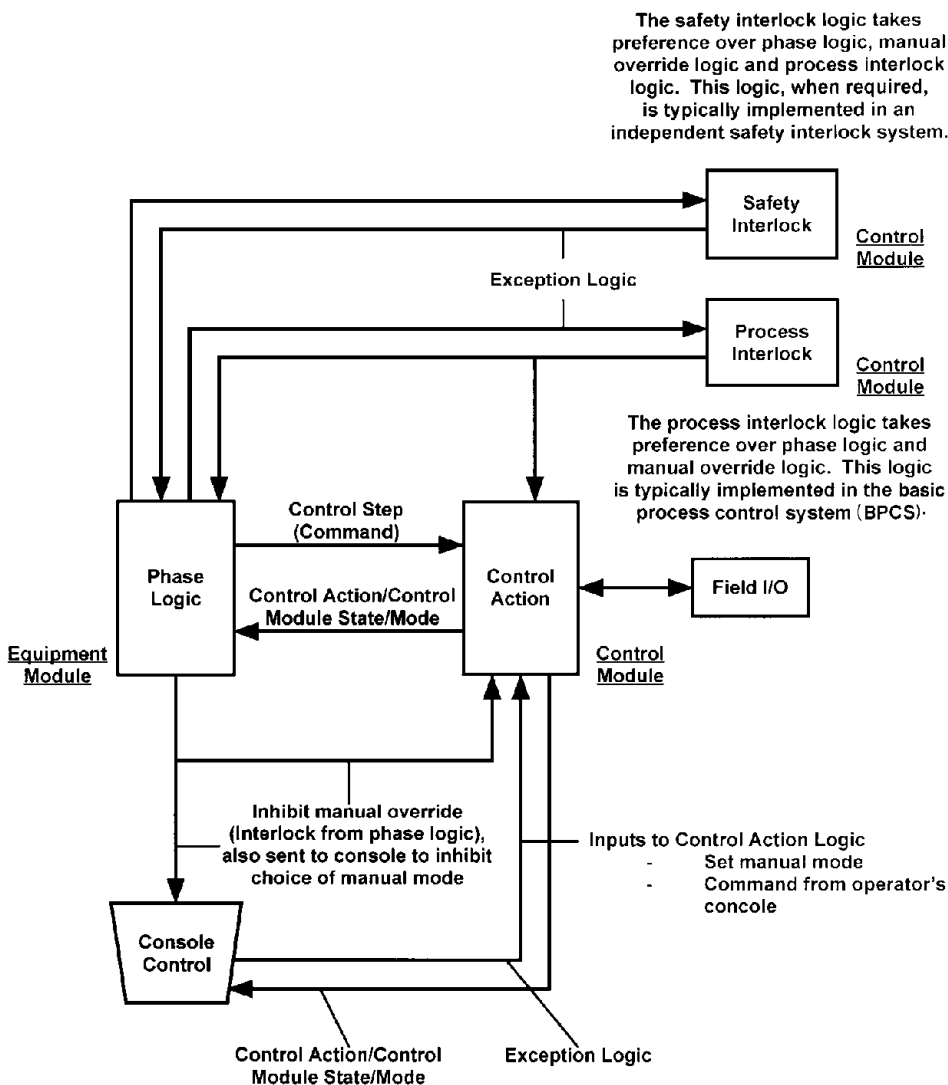


FIGURE 16 Exception handling in control actions.

## Key Points

1. The S88.01 standard is ideally suited to a methodology for the design of batch control that is object oriented.
2. Logic that changes frequently should be incorporated in the recipe procedure, when possible.
3. Logic that changes infrequently should usually be incorporated in equipment control.
4. Partitioning of process cells requires knowledge of both the equipment in the process cell and the recipes for the products that will be made in the process cell.
5. Exception handling can amount to 50–80% of the total configuration time for a batch project.

6. Reusable software components provide a lot of advantages in a batch control system.
7. Data collection is part of each equipment entity.

### Rules of Thumb [6]

1. Understand the process before designing a batch control system for that process.
2. Design the control system before beginning implementation of the control system.
3. Get the end users involved in the design.
4. Exception handling is usually best done in equipment control.
5. The use of states simplifies dealing with exception handling in equipment control.

### REFERENCES

---

1. ANSI/ISA-S88.01-1995, *Batch Control—Part 1: Models and Terminology*, 1995.
2. IEC61512:1997, *Batch Control—Part 1: Models and Terminology*, 1997.
3. NAMUR NE-33, *Requirements To Be Met By Systems for Recipe-Based Operations*, May 19, 1992.
4. Hopkinson, P., and J. Hancock, "A Case History of the Implementation of An S88-Aware Batch Control System," *World Batch Forum*, 1998.
5. Fleming, D. W., and P. E. Schreiber, "Batch Processing Design Example," *World Batch Forum*, 1998.
6. Christie, D. A., "A Methodology for Batch Control Implementation—Real World Lessons," *World Batch Forum*, 1998.