Safety Logic in Modular Batch Automation

Thomas E. Crowl
Principal Application Engineer
Siemens Moore Process Automation Inc.
1201 Sumneytown Pike
Spring House, PA 19477
USA
Tel (215) 646-7400
Fax (215) 283-2801
tec@moore-solutions.com

Cynthia L. Benedict
Lead Project Engineer
Siemens Moore Process Automation Inc.
1201 Sumneytown Pike
Spring House, PA 19477
USA
Tel (215) 646-7400
Fax (215) 283-2804
benedictcl@moore-solutions.com

KEY WORDS
Batch, Exception Conditions, Safety Logic, Safety Instrumented Systems, Safety PLC, ISA S88.01, IEC61512-1, ISA S84.01, IEC61508-1

ABSTRACT
In the early days of batch automation there was usually a central computer that controlled everything. This computer ran recipes, executed sequential logic, did data acquisition of process variables and also performed direct digital control (DDC) of analog and discrete devices. Since one computer did every thing from sequencing to DDC it was only natural to imbed the shutdown and safety logic into the batch sequential code that was running normal operations. And since one huge monolithic program ran the entire process, the safety logic was always running. In modern S88 (IEC61512) based modular batch automation systems the monolithic code has been replaced by smaller reusable phases controlled by a batch manager that runs recipes. Many who have grown up with DDC imbed safety logic inside the phases. This approach requires an active equipment phase at all times to keep safety logic available at all times. There is a problem with this approach. Phases are transient by nature. They have a beginning and an end. You cannot guarantee that there will always be an active equipment phase. Although there may be some holding logic associated only with a specific phase, often this logic is generic and should be moved up to the unit level. This paper looks at methods available to the user for safety and exception recovery logic in current modular batch systems. Included are case studies of five separate batch projects where recognizing exception conditions and executing safety shutdown logic was essential.
INTRODUCTION

This paper looks at the implementation of safety and exception logic from a batch automation engineer’s perspective. The concentration is on the overall project requirements, in particular the demands on the batch system, rather than focusing on the detailed safety logic. However some safety industry terms are used throughout and must be understood to read this paper. Detailed explanation of these terms can be found in ISA S84.01 (IEC-61508-1) “Application of Safety Instrumented Systems for the Process Industries.” Automated Safety Instrumented Systems (SIS) can be electro-mechanical relays, motor driven timers, solid-state logic, relays, and timers, hardwired logic, Programmable Electronic Systems (PES), or any combination of these. Today many SIS are centered on Programmable Electronic Systems (PES) designed to replace much of the traditional hardwired safety logic. In most cases these programmable electronic based systems can be as reliable and easier to maintain than hardwired systems. A standard PLC can be used with special circuits and application level programming or a safety PLC can be used to implement a PES. Safety PLC’s are pre-engineered to meet a Safety Integrity Level (SIL) and are usually certified by an agency such as TÜV. SIL denotes the level of safety performance for the SIS. The SIL can be 1, 2, or 3. The higher the number, the higher the availability of the safety function is. SIL 2 is equivalent to TÜV AK4 and SIL 3 is equivalent to TÜV AK6. The Basic Process Control System (BPCS) is denoted as being separate from the SIS and is usually handled by a Distributed Control System (DCS) or PLC.

For examples we look at five separate projects. These five projects were selected for the wide range of exception and safety concerns which they cover. This range starts from a pharmaceutical manufacturer that has few traditional safety concerns; the paramount concern in this process is quality. The second is a chemical manufacturer. Although there are some safety concerns, the process is a relatively safe one. There is extensive interlocking, but most of the interlocks are for product quality and equipment protection. The third process has more significant safety concerns. A failure in this plant could result in a hazardous situation. This is the first case study where a safety PLC SIS was used. The fourth project is similar to the third with the same end product yet an even more dangerous process. A failure here has the potential destroy the entire plant site. Again an SIS was implemented with a safety PLC for this process. The final process uses the same material that caused more than 2000 deaths in Bhopal, India in 1984. Here a failure not only threatens the local plant site but the entire population downwind of the plant. A safety PLC was also used to implement the SIS for this process. This is not a paper on process hazards review, failure mode analysis and SIS. It instead concentrates on some of the concepts and tactics used in safety systems used with batch and how some of those same tactics can (and should) be used in general equipment and quality interlocks or exception conditions. The paper analyzes this range of processes and the solutions to their specific applications. It looks at the similarities in the solutions and shows where techniques used at one end of the safety spectrum can be used at the other end. The paper will also show that even though safety concerns are always the top priority, the quality of the product made, production quantities and protection of expensive equipment is as important in a process that requires a safety system as it is in a less hazardous process.

CASE STUDY 1

This is a biotech fermentation and purification process. In this process safety concerns are minimal. The operating temperatures are approximately body temperature. The operating pressures are around 2psi. Other than the cleaning solutions and agitator blades you could swim in most of the vessels with no ill effects to you. Safety concerns are limited to over temperature, over pressure and the possibility of open
connections on a transfer or cleaning operation. These were handled with software interlocks in the DCS. A larger quantity of interlocks and exception logic in this job was focused on product quality and equipment protection. In pharmaceutical manufacturing product quality translates to patient safety. In addition to the safety factor, a damaged piece of equipment may cost ten thousand dollars and a contaminated batch may mean the loss of over a million dollars. The only limit switches in the system were on valves that went to drain (because you don’t want to accidentally send product to drain) and there were proximity switches on the manual transfer panel connections. The batch manager enabled the proximity switch checking and alarming after the recipe prompted the operator to make the connection and the operator acknowledged that the connection was made. A proximity switch alarm would notify the units on either end of the transfer panel connection and the units would take appropriate action. The original design of this plant placed most of the exception logic into phases. Because of the speed required by the process and the shock to the associated equipment some pumps were left running between phases in the CIP and chromatography areas. Each of the phases had logic to shutdown the pumps and associated valves when they went into hold. The hold could be triggered by process conditions or by the operator. It was discovered during debug that the possibility of a hold being triggered when there was no active phase was higher than expected. With 20 to 30 recipes being run daily, a batch would be held with no active phases running about once per week. When this happened the pumps would be left running. To avoid this situation it was decided to move the holding logic out of the phases up into the unit. The unit recognizes the hold condition of the batch and runs its hold logic. Since this portion of the hold logic is the same for all phases, moving the hold logic up to the unit meant that each individual phase required less hold logic and memory usage was reduced significantly in the controller. This philosophy was also used in the bioreactor units. Cell culture fermentation is a slow operation and most of the operation is at a steady state. It was decided to only use phases to change the state of the bioreactor and not have a cell culture phase that runs for the sole purpose of providing exception logic. The exception logic for steady state operations was at the unit level. This meant that for over 90% of the time in the cell culture operation there were no phases running. For this process the 90% translates to weeks of each batch running with no active phases. Allocated units are the source of the status of equipment and the batch not the phases. Phases come and go but unit supervisory logic is always there.

CASE STUDY 2

The process in this application is the manufacture of bulk polystyrene beads. Final use of the beads ranges from cups and plates to high tech structural composites. This product is made on a small margin, turnaround times, cycle times, and production quantities are critical to a profitable operation. The plant was automated using a DCS with the batch management handled by a S88 based batch management system to increase the throughput of the facility without additional vessels. In the process the polystyrene beads are formed in a reactor and then pumped to the drying section of the plant as aqueous slurry. The batch automation system covered raw materials, reaction, drying, and finishing. Bagging and palletization of the beads is handled by a separate PLC. There is no programmable electronic safety system. Critical safety interlocks are hardwired. All of this hardwired interlocking is repeated in the DCS. The entire process is heavily interlocked in the DCS. These interlocks are at both the control module and phase level. Any interlock at the control module level will then hold the phase that controls it. In addition each valve has open and close limit switches that the phases use for transition logic. The critical processing steps in this process center around the reaction operation, which originally was designed to be three phases. The plant had previously been controlled by a DDC system and these three phases corresponded to subroutines in the DDC system. Once the reaction is started, timing is critical. If the reaction is allowed to proceed too far, the beads will be too large to pump or the reactor may solidify completely. If the reactor solidifies
there is a messy and time consuming clean up. It is not possible to simply stop the reaction on a trip. The reaction will continue regardless. Although all three phases had logic to recover from trips and actions to allow for the reaction to continue, or slow the reaction, or stop it, there was a possibility that a failure would trip when there was no exception processing running. Even in a recipe with no transitions, with each phase following the previous one, as one phase ends and the next begins there can be an instant where there are no phases running. This had the possibility of adding time and effecting processing. Since the all products on this line required these three phases to run in the same sequence, it was decided to combine the three phases into a larger reaction operation that was programmed as a single phase. This way the entire reaction sequence could run to completion with no interaction from the batch manager. With a single larger phase the complex recognition of an exception condition and the resulting hold, recovery, and restart logic could remain imbedded in that phase. The drying section of the plant had two phases, a startup and a shutdown phase. These phases sequenced the heavy equipment on in a manner designed to prevent electrical substation trips and to prevent damage to the equipment. These were initiated by the operator and not by the batch manager. The shutdown phase could be triggered automatically from the unit level on an equipment trip.

CASE STUDY 3

The process for this plant is an emulsion process for the manufacture of latex resins. In this plant the design divided the interlocks into three categories. Quality interlocks to protect product. Equipment interlocks to protect equipment. Safety interlocks to protect personnel. The implementation used two systems, a DCS to control the batch operations and a safety PLC for the critical safety interlocks. The quality and equipment interlocks were programmed in the DCS. There is a maintenance bypass that can be used to bypass quality and equipment interlocks but it cannot bypass the safety interlocks. This process is on the borderline for a requiring an SIS. Although other manufacturers may run this type of a process without a safety system, company guidelines required that this plant have an SIS. All critical valves, pumps, and limit switches were wired into the safety PLC. The safety PLC was continuous logic that ran whether there was a batch in the reactor or not. The safety PLC and the DCS had outputs wired in series on this application so that the DCS would be able to open and close valves needed for normal batch operation but the safety PLC would have final say on any valve position in case of a trip. There was no double wiring of the inputs. All inputs wired to the safety system were also brought into the DCS via packed words across a dedicated communications link. All interlocks in the safety system were mirrored in the DCS. Some of the limit switches of the valves controlled by the DCS are wired into the safety PLC and this required a longer than typical delay to prevent a disagree trip on the valve. This longer delay allowed for the additional communications time required getting the value from the safety PLC. On this process there is only limited interaction between the batch control and the safety PLC. The batch and any running phases are placed into hold if there is a trip on the safety PLC. The safety system looks at the recipe being run and sets temperature limits based on a look-up table it has in its memory. If it does not see a valid recipe name, it sets the temperature limit to a default temperature limit. This allows the end user to run a chemical clean recipe or run a hotter product recipe without tripping the safety system. It also allows the user to run the safety PLC with much tighter temperature limits than they could have with a generic temperature limit. There are a few points during normal batch execution where a trip of the safety system is common. In these places exception logic was built into the phase in the DCS to take action to clear the interlock condition and then automatically reset the trip in the safety system without operator intervention, allowing the automatic system to seamlessly control the process through the trip.
CASE STUDY 4

This application is on a latex emulsion batch process where the reactor is blanketed in high pressure (over 1000 psi) Ethylene. Although there are little safety concerns beyond the plant boundaries, the explosion potential meant that this project would require a SIL 3 SIS. A safety PLC certified to TÜV level AK6 was used to implement the SIS. All critical inputs, valves, and pumps were wired to the safety PLC. The system used an Excel spreadsheet based cause and effects matrix to program and view the safety system logic. The DCS handled all the other process inputs and outputs. The DCS and safety PLC were wired separately and there were no signals wired to both systems or between the systems. The DCS based batch interlocks mirrored the safety PLC interlocks. The safety system outputs were used as an interlock to the DCS valves and motors. This allowed fewer signals to be sent between the two systems and less double interlock programming than if the DCS brought over each individual interlock and wired each condition to the control module in the DCS. Because this approach only looks at the output to the valve or pump a second variable was needed that showed if all the interlocks were satisfied and the output could be energized if it needed to be. The communications between the DCS and safety PLC were over a dedicated communication link and all discretes were packed in words. One of the concepts stressed on this application was the elimination of double entries. If a trip point is entered into the safety PLC that same trip point should not be reentered into the DCS. If it is, there is a possibility of error and if the safety PLC is changed the DCS must also be changed. The safety PLC trip point should not be copied into the DCS but instead the variable should be referenced by the DCS. The dedicated communications link helped with maintaining this goal. Since the safety system had all the critical analog values and limit switches wired into it, a second cause and effects matrix was created for level 2 interlocks, which were not as critical. By placing these into the safety system the number of communication points was reduced and the user could exploit the trouble shooting capabilities the cause and effects matrix brings to complicated interlock logic. This application used batch oriented hold and recovery logic that was used to recover from a safety system trip in an orderly fashion. This hold logic resided at the unit level and would automatically start based upon one of 4 or 5 trips from the safety PLC. The main task of this unit hold logic was to get the DCS in line with what the safety PLC had done and to clean up after the safety PLC emergency shutdown. Once the process is properly shutdown and stable, the DCS based recovery logic could be started. The recovery logic resets the safety PLC and restarts the process in an orderly fashion. Once the recovery is complete, the regular recipe operations can then be restarted. These hold and restart tasks are equivalent to phases although they were not specified as phases in project documentation. These hold and restart tasks are not safety related but product quality and equipment protection related. The ability to trip the hold logic is enabled at a specific point in the batch and then disabled later in the batch after the need for the hold logic had expired. Running a phase in the recipe enables and disables this logic. Using a phase to enable the logic allows the point at which checking begins and ends to be changed from product to product.

CASE STUDY 5

This process is a single product single train process with the choice of two storage units at the end of the train. The process at this plant is similar and uses the same raw materials as the infamous Bhopal, India plant. Even though the process has been redesigned to reduce the risk to the surrounding public there is still significant safety concerns with making this product. This plant is in the United States and the surrounding population has grown significantly in recent years. Due to the critical safety concerns the application used a safety PLC system TÜV certified to AK6. Due to the batch requirements on the job it also used a DCS with S88 batch management supervisory control. The implementation was designed with control, data acquisition and alarm functions separated into three categories, batch, interlock, and safety.
Valves, limit switches, and analog inputs into the system were designated as batch, interlock, safety, or some combination of the three. In addition to process warnings and alarms the critical safety system alarms are designated as HHH or LLL alarms. All safety functions are handled by the safety PLC system. Batch and interlock functions are handled by the DCS. The safety application used a large excel based cause and effects matrix to aid in the design, configuration, and operation of the safety system. All valves had limit switches that were wired into one of the two systems. The safety system and the batch control operated separately but had communications between them. Anything safety related was wired into the safety system yet many of those signals were also needed for the batch control. It was decided to get these signals over a dedicated communications link between the safety PLC and DCS rather than to hard wire the signals. In example, the limit switch for a batch valve may be wired into the safety system yet the output to the valve is controlled by the DCS. The status of this limit switch is needed for batch execution yet it is critical that the safety system know the status of this valve. Boolean values are packed into words. Analogs are sent in arrays. The safety PLC could interlock phase execution but phase execution had no effect on the safety system. The only way the safety system was effected by the batch system was that the batch system may open valves that the safety system may recognize the limit switch as a causal effect and therefore shut down since the proper alignment of valves was not met. SIS is active continuously and will act to keep the plant in a safe state even when no batches are being run.

REVIEW

In this paper the control systems for five separate plants were summarized. Three of these plants required formal SIS. This review looks at the similarities between the three applications with SIS first and then moves on to the other two processes.

All of the SIS case studies segregated the safety logic from the rest of the application. A basic premise of safety systems is to separate the safety logic from the Basic Process Control System (BPCS). In all of the safety applications, the safety system is physically separate from the batch system. In only one does the safety system even glean information from the DCS. The safety logic runs continuously always checking the status of the equipment looking for a situation that could be hazardous. They do not rely on the batch to tell them when there should be a dangerous situation they look at the physical state of the plant to make that determination. The safety checks are always active, the system can modify limits based on the recipe but the batch does not turn on the checking. Trips in the safety system will cause hold logic to be executed in running phases and will often cause the unit to initiate recovery phases. Although it is easy to lump these applications into a safety category, quality and equipment concerns are as important in these processes as they are in any other process. In other plants there is often a tendency to lump safety, equipment, and quality into one amorphous bag. In all three SIS case studies there was a clear separation between safety, equipment, and quality. The same tenacity and rigorous approach that was applied to safety design tends to follow through with the equipment and quality designs. This separation and rigorous approach to quality can be applied in other process industries where hazards are not as critical, thereby raising the quality awareness to the level that hazards have in the safety industry.

In the other two case studies there are some similarities and differences from the applications that used a SIS. In all cases the units had the ability to start shutdown phases based on trips. In all cases batch logic had the capability of enabling and disabling continuous logic that looked for exception conditions and acted on them. Where they differed was in the tendency of the non-SIS projects to intertwine quality and equipment logic into phase code. Although this had no effect on most of the plant it caused potential problems in each of the applications. The potential problems were due to applying the concepts of
monolithic batch to modular batch and a failure to segregate and analyze all possible modes of failure for equipment and quality. The potential problems that were observed are examples of the pitfalls of modern modular batch automation. Any time that there is a server that runs the batch manager software and the phases are in a different hardware box, there is the likelihood that there will be periods of time that there are no active phases. The potential problems were solved in two different ways. In case study 2 the phase’s size was increased so that all exception logic associated with the reaction operation could be imbedded inside of the phase. In case study 1 the exception logic was moved up to the continuously running unit level were it could always be available. Case study 1 takes the approach a step further by using phases that change the state of the unit then get out of the way. The exception logic is handled at the unit level. The end user has found that most operators consider no active equipment phases to be a good thing. To them it means the system is stable and they only need to worry about basic control, and the setpoint changes that they make themselves.

**SUMMARY**

The decision to have a safety system is mostly an economic one. The cost of having an accident far outweighs the cost of preventing one. With interlocks and alarms for equipment and quality the case is also an economic one. Equipment is interlocked because it is expensive to replace and may end up costing even more in down time and lost production than the cost of equipment. Quality is important across the spectrum not only in the pharmaceutical industry where peoples lives are ultimately at stake but also in commodity chemicals where quality can be the difference between product becoming a beer cooler or a motorcycle helmet. A premium quality can command a premium price.

The separation and rigorous approach that is applied to safety can be applied to quality, thereby raising the quality awareness to the level that hazards have in the safety industry. For the same reasons that safety code is segregated from basic process control, quality and equipment interlocks should be continuous and segregated from batch control.

It is best to design application code assuming that there may not be an active equipment phase to handle exception logic. There is some phase specific exception logic, but exception logic that is not phase specific is the domain of the unit. This exception logic should run continuously in an allocated unit, not in a phase that starts and ends. The execution of phases can set limits and enable and disable this logic. If a phase has completed its task, it should not be left sitting around waiting to run exception logic.
BIBLIOGRAPHY