Session Nine

How could it be considered "good engineering practice" to bypass your SIS during the most critical time of your process?

Luis M. Garcia G. CFSE
Siemens Energy & Automation, Houston, Texas

Key Words
Process Safety, Plant Start Up, BMS (Burner Management System), Permissive Sequencing, Safety Availability, Reliability, Cause and Effect Diagram, Safe Charts.

Abstract
Although most facilities embrace ANSI/ISA 84.00.01-2004 (IEC 61511) and the Safety Life Cycle (SLC) as the way to comply with regulatory requirements (Like OSHA 1910.119), there are specific instances when most operations deviate from the standard. These are during start-up, shut-downs and process transitions. Processes with adequately designed Safety Instrumented Functions (SIF) that are validated to well developed Safety Requirement Specifications (SRS) are commonly (although momentarily) idled, and instead are practically replaced by a team of operators, managers and specialized personnel. Bypassing, inhibiting or masking is a common practice during these plant conditions. In these cases, the Safety Instrumented System (SIS) is temporarily replaced by humans in calculated and intensely watched conditions.

This paper questions the need for this practice and confronts the practical limitations that lead to it. It examines the assumptions used to justify the suspension of certain SIFs and uses Burner Management Standards and typical process SIS, as an example of how to automate the permissive sequencing required for these process change of states.

Finally, the paper examines how a cause and effect tool could be used to simplify the development and implementation of automated permissive sequences including verification and validation as required in the standard.

Table of Contents

Introduction

There has been wide scale adoption of the functional safety concepts in the process industries as a way to deal with process risks and to control the safe operation of facilities. In particular, the S-84.00.01 - 2004 (IEC 61511 Mod.) standard has become recognized as a fundamental definition of how to implement a Safety Life Cycle and design of Safety Instrumented Systems (SIS) for the process industries. However, implementations have been constrained to steady state protection functions and rarely applied to sequencing, either during start up, shut down or dynamic transitions. Sequencing has almost always been left to a manual procedure and operator's discretion.

Start-up, shutdowns and transitions have always been considered the most dangerous period for operations in the process industries. If that is the case, what is the reasoning behind the suspension of Safety Systems during those periods and is that reasoning justified? In addition, do improvements in technology offer new ways to address some of the assumptions about permissive sequencing?

Permissive Sequences

Armed with a full set of steady state operating conditions and a list of process constraints, the SIS system is designed to offer a layer of protection above the Basic Process Control System (BPCS) and the operations team. While designed well to protect the process at steady state conditions; getting to the steady state typically involves a permissive sequence. Bypassing, inhibiting or masking is a common practice during these plant conditions; in these cases the Safety Instrumented System (SIS) is temporarily suspended.

In order to understand the reasons behind such a limiting practice on the use of Safety Systems we must understand first what is involved in the implementation of a permissive:

Permissive Sequences have three general characteristics:

- A time dependency that must be considered
- Changing variable thresholds or limits
- Interlocks that vary or may need to be inhibited or overridden

Assumptions for Suspension of SIS

There are five key assumptions that are used to explain and justify the suspension of Safety Instrumented Functions (SIFs) during process transitions:

1. Processes transitions (i.e. start-ups), are not frequent and are of short duration compared to steady state operation. Therefore, SIFs can be suspended and Start- Up carried out manually with a written procedure under the supervision of a Start-Up Manager.

2. There is a lack of similarity between different processes. This makes prescriptive standards impossible and best practices difficult. Therefore, it seems acceptable to manage them manually under tailored conditions.[1]

3. There is a lack of similarity between the Process Transition operation and Steady State operation. Safety Systems are therefore designed to operate under Steady State conditions where the majority of the operating time occurs. SIS designers would have to create an entirely new and conflicting SIS to manage process transitions.

4. The process transition operation is more affected by operational subjectivity and procedures than steady state operation, i.e. "How long an interlock should be bypassed?" Therefore automating process transitions require strong Operations input in the development process.

5. Because the transition is sequential and dynamic, timing of process steps and interlock changes are critical. These are difficult to validate and verify without both detailed operational knowledge and adequate (proper) simulation routines.

Assumptions Challenged

While these assumptions may seem valid at first glance and certainly are expedient, a closer examination in light of fundamental process safety concepts proves otherwise.

1. <u>Process transitions (i.e. start-ups), are not frequent and are of short duration</u>

   Process transitions, represent the most volatile time for the process. The variable can change significantly and the Basic Process Control System (BPCS) may not be capable or tuned to handle the process movement. This is a dangerous time to leave it all in the operator's hands because of the amount of other things they are required to monitor and execute. The complexity of the transition process (timing, changing thresholds) requires the operator's full attention. Asking the operator to provide an additional Safety Protection Layer on top of that focus will increase the level of risk and can be dangerous.

   The human factor has been recognized to severely limit the dependability of

the risk reduction factor. A Layer of Protection must be dependable and auditable. Neither of these characteristics would seem to apply to a bypass situation. During process transitions, variables are changing rapidly and protection thresholds are also subject to change. It is not the time to depend on a less reliable protection layer.

---

[1] Exceptions to this are common applications like BMS, where similarities have allowed for more prescriptive standards as for example NFPA 85.

2. <u>There is a lack of similarity between different processes.</u>

   While the lack of similarity between processes does increase the difficulty of using Safety Instrumented Systems, it does not remove the responsibility for insuring the safe operation of the process at all times. If it is difficult to automate why would we expect that the operator is going to find it easier to make the right decisions during a complicated transition? In fact, the very lack of similarity between processes is a reason to work out the transition in advance and to make sure the safety systems remain in effect.

   However, there are similarities in the control strategies for different processes and we will show that there are ways to deal with them in a consistent manner.

3. <u>There is a lack of similarity between the Process Transition operation and Steady State operation.</u>

   While in many processes, the majority of time is spent at steady state, the more dangerous times are during transitions when variables are changing rapidly and the process is in conditions that the BPCS was not designed to handle. I.e. controller tuning may not be adequate for loops during transitional period. What we are really challenging is the practices of letting the operator do it because it is "difficult" to create an SIS that would handle transitions. (an exception to this has been those applications where strict prescriptive standards applied, like for example NFPA 85 & 86).

   If we do our job correctly, the time spent on writing and properly training operators in a seldom used start-up procedure could be better spent on properly designing the SIS system to handle transition routines. The properly engineered SIS should consistently outperform a stressed operations staff. We will show later that using advances in programming technology, it is possible to simplify the design and validation.

4. <u>The process transition operation is more affected by operational subjectivity and procedures than steady state operation</u>

   Again, we are allowing "difficult" as an excuse to give up on safety. In reality, the same level of operational input is required to write the procedures needed for a transition routine as to write an automated SIS. There are two real

difficulties for getting the proper operations input.

---

[2] Human Error (action or inaction) as defined by ANSI/ISA 84.00.01 (part 1) or IEC 61511-1 Mod. Definitions - 3.2.32 page 26 Note: ANSI/ISA 84.00.01 Part 2 or IEC 61511-2 Mod Offers guidance on how to include operator's availability and reliability calculations.

First is the sequence of project steps, i.e. it is difficult to get operational input at the software design phase but less difficult at the procedure writing stage. To do it right, operations must be involved throughout the project.

The second issue is the lack of communication tools between the operations group and the software design group. It is not easy to translate the needs of process operations into usable SIS code.

5. Because the transition is sequential and dynamic, timing of process steps and interlock changes are critical.

   The dynamic behavior of the process is the very reason that the process should be automated. It does require a robust simulation routine with the participation of process and operations personnel. However, the idea that we leave that to a written procedure reduces the dependability of an independent protection layer. Since simulation is very difficult with a manual procedure, automation, with proper simulation tools, is the better answer.

Sequence Requirements

Two things are required to adequately define and automate Permissive Sequences:

- Thorough knowledge of the process and its operation
- A set of tools to handle dynamic safety logic.

Operations Knowledge in the Project Sequence

In the design of SIS Systems, Operations traditionally have been involved in the early stages for the Process Hazard Analysis (PHA) and again during Design Review to insure the operational capability of the final design. Operations are then given the completed unit to start-up. Therefore, the bulk of the design data is based on the process information which traditionally has been at steady state conditions. To automate the Safety Functions during critical process transitions, significant Operations input is required along with the basic process data during the software design stage.

It is difficult to get Operations time on an ongoing basis. In addition, the Operations group and the software design team come from different backgrounds and use different terminology, making it more difficult to communicate the needs of the software design team effectively. Anyone who

has gone through design review with Operations with stacks of ladder logic diagrams will understand the challenge.

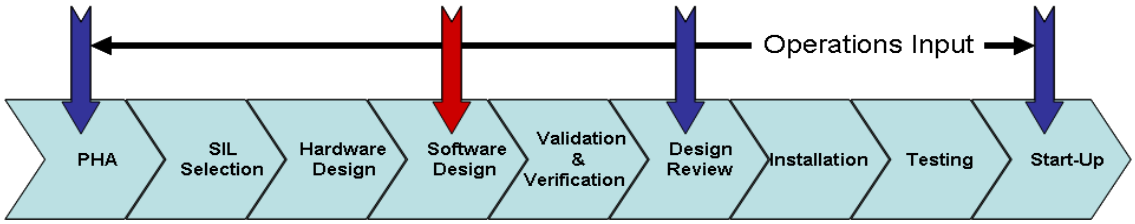

Figure 1: Operations Input into SIS Design

However, if the process safeguards are to remain intact during process transitions, it is essential to understand the process that Operations will follow and to understand what is practical to expect in the real world. If the safeguards are not implemented in a "reasonable" manner, it is likely that they may be bypassed during actual operation. Therefore, one of the first decisions to make is to find a common language of communication between the Operational and Engineering personnel.

A traditional way of looking at Process Shutdown Logic has been with a Cause and Effect Diagram. The Cause & Effects matrix was originally derived from Safe Charts in API RP 14C for offshore platforms and is commonly used in the process safety industry for documenting safety requirements[3]. In a cause and effects diagram, a set of process deviations, or causes, is listed in rows down the left side and a set of process responses, or effects, are listed in columns across the top. The intersection cell in the matrix defines the relationship between the cause and the effect.[4]

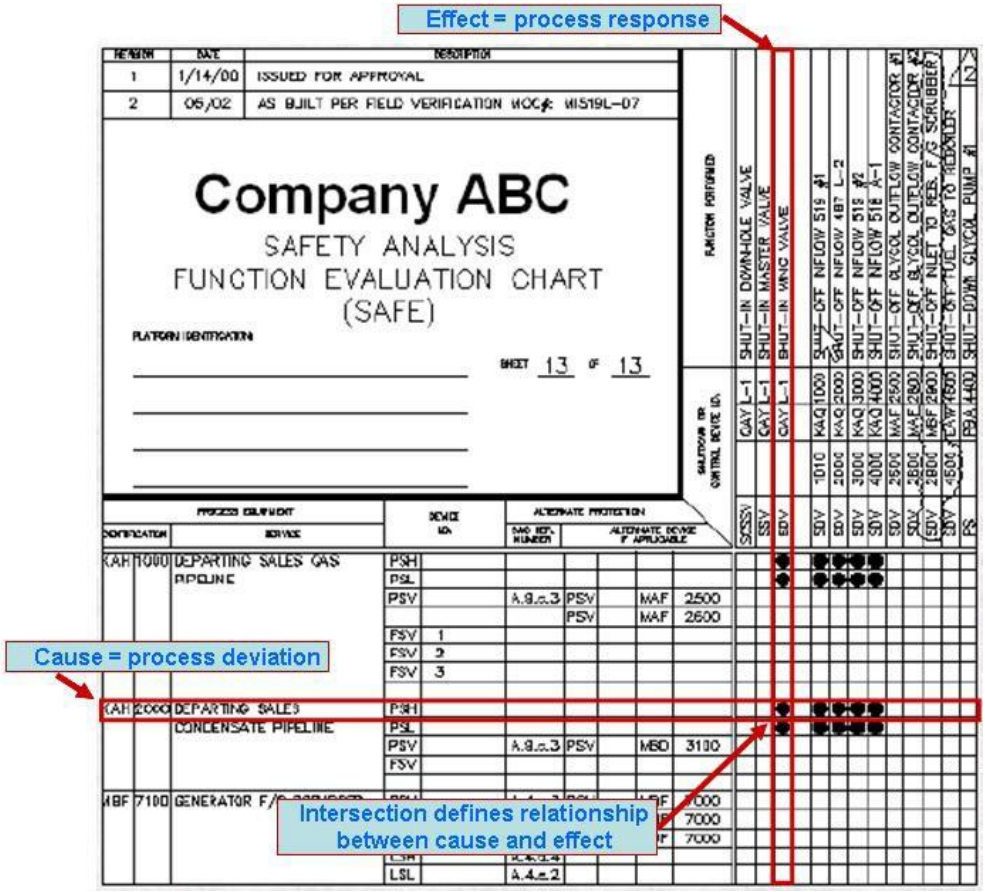

Figure 2: Cause and Effect Diagram

[3] Assigning Safety Requirement Specifications (SRS) to specific Safety Instrumented Functions (SIF)

[4] These central panels might have intersections that would light up, relating active causes or anomalies in the process with active effects or process protection.

The cause and effect diagram has become very popular amongst Process Safety Professionals because it is an easy method to bridge the communication gap in the SIS design team. The diagram is an easy way for those familiar with the process and operations to understand the logic being implemented in the Safety System. Once the cause and effect relationships have been agreed to, they can be translated into the Safety System program.

A major limitation of the Cause and Effect Diagram has been the ability to handle the type of dynamic safety logic that is seen during a process transition. Permissive sequencing is difficult to portray in a static matrix. The matrix, originally designed to relate causes to effects with simple intersections (lights that indicated active causes affecting active effects as shown in Figure 2), needed more options when defining these intersections, not only to make possible dynamic logic, but to generate comprehensive validation reports.

Tools for Dynamic Logic

There are three major characteristics a configuration tool must have in order to be able to handle changing logic. These are:

1. Override
   s
   - Control Overrides as Function of Process Variable (causes)
   - Set up Permissive Timing (see Time Dependency)
2. Variable
   Thresholds
   - Control relationship between Process Variables (cause) and Process Reactions (effects)
3. Time
   Dependency
   - Definition of Steps
   - Limit of overrides
   - Control of Step Length (Delay, Prolong)

Time Dependency

In a cause and effect environment, the time relationship between the cause and the corresponding effect can take four forms (Figure 3)

To understand a time dependent step, let's consider the purge of a furnace. If the flow rate is a constant, then the way to assure complete purge is waiting until sufficient volume of air sweeps through its hearth.

In this case, the process variable is time and a delay post trip will not allow the next step until after the configured duration of the process.

Therefore one should consider four types of time dependency (see Figure 3):

1. No Time Function
   The Effect occurs as soon as the Cause is active

2. Pre Trip Delay or ON delay
   The Effect occurs a timed duration after the Cause is active

3. Post Trip Delay or OFF delay
   Effect is active for a timed duration after the cause is cleared

4. Timed Cause
   The Cause is active for a timed duration after it is triggered regardless of status

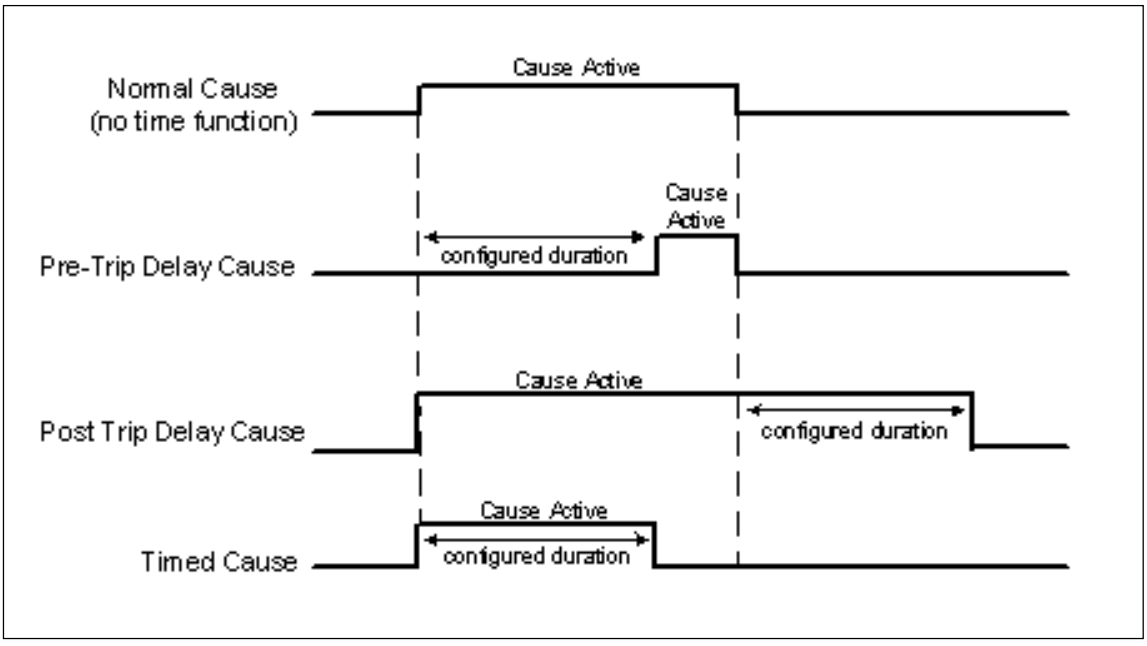

Figure 3: Timing Diagram for Cause Time Functions

Variable Threshold

Purge in a BMS application, is a good example of variable threshold.

Purge must be performed at a predetermined Air flow rate which is usually much

higher than the one required for optimum combustion (fuel to air ratio). Therefore the air flow rate, after purging is completed, must be lowered before lighting the pilot or the burners, without aborting the sequence. This can be achieved defining the relationship between different triggering points for the same variable (cause) and selectively defining their relationship with the process reaction or effect. This is done using normal or "N" intersections and resettable-override or "R"[5] intersections as appropriate. (See later)

Control Overrides

Dynamic logic requires that an effect is able to be overridden independently of some causes. For example, in a furnace, we want to trip the furnace when we lose flame so our static matrix shows a flame cause and a fuel valve set (Double Block and Bleed) effect. However, on start-up, we need to be able to open (override) the fuel valve set to ignite the burner.

In addition, we have to be able to not allow an override based on other causes such as pilot flame.

Intersections of the type "resettable-override" allows for a process reaction to take place (or effects) despite process variable (or causes).

These overrides are time constrained and could only be applied if there is no active process variable (cause), with a normal "N" intersection related this particular effect, allowing for sequence conditioning.

This is, for example, the case of the set of double block and bleeds valves which define the SIF of a burner. If there is a loss of flame, sensors won't detect flame (variable or cause), and then the set of valves will block the gas. To light the burner, the action of the flame sensors must be temporally overridden, and this is done with the "R" intersection. On the other hand, the override cannot be allowed if there is no flame in the pilot, and therefore the intersection of the cause "pilot Flame" should be of the normal type. In some instances the sequence might involve turning the pilot flame off after the main burner is on and this could add complexity to the process. In such case, a new cause that reflects "flame in the pilot" while pilot valves are open should be created with a normal intersection to the main burner set of valves with a delay "post trip" to allow transition.

Finally, the duration of an override is another critical point to take into consideration. An override can not last indefinitely. In the case of Purge for example, the time in which next step (light the pilot) should be allowed after purge is completed should be well assessed during engineering phase.

A Dynamic Cause and Effect Matrix

It could be concluded then that for a cause and Effect matrix to be an efficient **configuring** and **documenting** tool that allows for **Validation** and **Verification** of an SIS's logic, during process **steady state** and **process transitions**, and following S84 standards; it would have the following characteristics:

[5]Resettable-Override (R) intersections combines the characteristics of both stored latched type S intersections that have to be reset once the cause disappears and type V Intersections where effect may be overridden manually by the operator. Effects connected to this intersection will remain latched when the associated cause becomes inactive, but may be overridden.

1 – Indicate active causes and effects independently of intersections. (For example: with the use of coloring of columns and rows – Red: Active, White: inactive, Green: reset etc.).

2 – Possibility of configuring (delaying and prolonging) when causes become active (as explain in "Time dependency")

3 – Possibility of defining different types of functions on how causes relate to effects (intersections); including independence to override, latch or complex voting causes architectures.

N: Normal (Effect will stay active while Cause is active)

S: Stored (Cause will trigger Effect until reset, regardless of inactivity of Cause - Latched)

V: With Override (Allows deactivation of Effect regardless of Cause)

R: Resettable Override (Same as V but with latch)

4 – Capability to time-limited overrides, and feedback Effect actions to Causes.

5 – Capability to dynamically simulate logic "off-line" to verify and validate configuration reporting.

An example of application

In order to illustrate the point let's consider a very simple example:

In this petrochemical process, a hydrocarbon gas needs to be dried. For such purposes the gas passes through a reactor packed with absorbent granules.

An exothermic reaction takes place in the "drier" allowing using temperature to evaluate its performance

If the temperature goes below certain level, (110 ºF), it is an indication that the granules are saturated and have lost their capacity for drying the gas. But because of thermal inertia, a 20 seconds delay must be allowed before the temperature is recognized as being too low.

On the other hand, humidity is extremely harmful for the process downstream,

and the SIF that protects the process has been ruled to be SIL 3 in a LOPA followed by a GAP analysis. (Figure 4)

Figure 5, on the other hand, shows how a cause and effect traditional "Static" matrix would be for this application. If four out of six temperatures go below 110 ºF, the unit will be taken to its safe condition, that is: Valves 110, 210, 130 and 230 will block, preventing the hydrocarbon from flowing downstream, while Valves 120 and 220 will allow any leakage recirculation. The "S" intersections indicates that the Effect will be "latched"

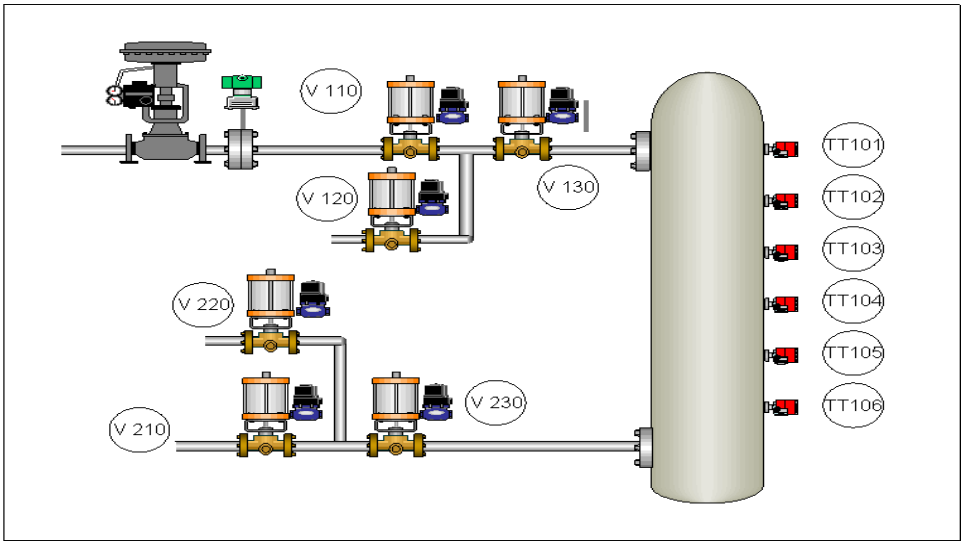

Figure 4: Example of application – Drying reactor

Company ABC **Safety Analysis** **Function Evaluation Chart** **Plant ID** **Sheet 1 of 20**	**Cause No**	**Valve 110 - Close**	**Valve 120 - Open**	**Valve 130 - Close**	**Valve 210 - Close**	**Valve 220 - Open**	**Valve 230 - Close**
Effect No		**1**	**2**	**3**	**4**	**5**	**6**
Temperature TT 101 < 110 °F	**1**	**4S**	**4S**	**4S**	**4S**	**4S**	**4S**
Temperature TT 102 < 110 °F	**2**	**4S**	**4S**	**4S**	**4S**	**4S**	**4S**
Temperature TT 103 < 110 °F	**3**	**4S**	**4S**	**4S**	**4S**	**4S**	**4S**
Temperature TT 104 < 110 °F	**4**	**4S**	**4S**	**4S**	**4S**	**4S**	**4S**
Temperature TT 105 < 110 °F	**5**	**4S**	**4S**	**4S**	**4S**	**4S**	**4S**
Temperature TT 106 < 110 °F	**6**	**4S**	**4S**	**4S**	**4S**	**4S**	**4S**

Figure 5: Static Cause and Effect Matrix for Drying Reactor

Let's now consider the start up sequence procedure as per the operational Manual:

Step No 1: Bypass all temperature sensors

Step No 2: Manually, open Valves V110, V130, 230 and V220, and keep Valves V210 and V120 closed.

Step No 3: From the BPCS, increase flow at a rate of 5 Gallons per minute every two minutes until reaching a stable flow of 30 Gallons per minute.

Step No 4: Once each sensor have been at a stable temperature above 110 ºF for at least 20 seconds, remove bypasses on sensors, one at a time, This should happen within the first 10 minutes of operation or system should be shutdown as packaging of granules are shown to be defective.

Step No 5: Ten seconds later, Open Valve V210 and Close Valve V220.

This is a complex operation that places a lot of pressure on the Operator's ability. Operators, at the same time are making decisions on alarms, process value, voting between process values, variables stability assessment, operational bypasses management and the most difficult of all decisions: Aborting if reactor does not behave as expected.

Let's now consider an automatic start up of this process, using a Cause and Effect Matrix with all five characteristics discussed above: (Figure 6)

Company ABC Safety Analysis Function Evaluation Chart		Timers	Max Override 10 Minutes	Max Override 10 Minutes	Max Override 10 Minutes	Max Override 10 Minutes	Delay Output 10 seconds	Delay Output 10 seconds
Plant ID		Override - Reset Tag	PB_START	PB_START	PB_START	PB_START		
Sheet 13 of 13		Cause No	Valve 110 - Close	Valve 120 - Open	Valve 130 - Close	Valve 230 - Close	Valve 220 - Open	Valve 210 - Close
Effect No	**Timers**		1	2	3	4	5	6
Temperature TT 101 > 110 °F	PTD 20 s	1	4R	4R	4R	4R	4N	4N
Temperature TT 102 > 110 °F	PTD 20 s	2	4R	4R	4R	4R	4N	4N
Temperature TT 103 > 110 °F	PTD 20 s	3	4R	4R	4R	4R	4N	4N
Temperature TT 104 > 110 °F	PTD 20 s	4	4R	4R	4R	4R	4N	4N
Temperature TT 105 > 110 °F	PTD 20 s	5	4R	4R	4R	4R	4N	4N
Temperature TT 106 > 110 oF	PTD 20 s	6	4R	4R	4R	4R	4N	4N
Effect No 1		7	R	R	R	R	N	N

Figure 6: Dynamic Cause and Effect Matrix for Drying Reactor

Notice all the information included in the dynamic matrix. From figure 6:

1 – Causes have a Post Trip Delay of 20 seconds, allowing for the stability claimed on the operation manual

2 – All intersections are of the type "R", for which all effect will be latched when triggered

3 – There is an Override-Reset TAG (**PB_START**) which could be connected to a push button and/or a switch with a key. (The arrangement should be Normally Closed to allow diagnostics)

4 – The Maximum time the override is allowed before aborting the process is 10 minutes, complying with what ids required by the Operation Manual. Therefore the reactor should be stable in 10 minutes or the system will shutdown and the process will have to be re-started.

If this program is implemented as indicated by the above dynamic matrix, the start up sequence, would be reduced to two simple steps:

Step No 1: Push PB_START

Step No 2: From the BPCS, increase flow at a rate of 5 Gallons per minute every two minutes until reaching a stable flow of 30 Gallons per minute. In fact this ramp which could be done automatically in the BPCS if the Safety System protection was in place.

The process will be protected at all times by the SIS, regardless of the operator's actions.

Conclusions and Recommendations

1 - Planning startup procedures for critical applications can be done with just a little more of engineering effort, at the beginning of the safety Life Cycle, when things can be easily changed.

2 - Unfortunately, for many critical applications, prescriptive standards do NOT exist that clearly define the proper sequence. Yet there are special applications, such as BMS, that clearly show how to do it. All one should do is to adopt a similar methodology based on controlled forced overrides limited by fully active Safety Instrumented Functions

3 - The benefits of allowing your SIS to stay in control 100% of the time during critical sequences (start up and shut down) are obvious.

4 - Performance based safety standards (i.e. S84) limit the amount of safety credit given to humans in a very wage way since it is very difficult to include "human state of mind" into the equations. Thus maximum avoidance should be recommended.

5 - Nowadays, there are easy to use safety rated programs (i.e. Safety Matrix) that allow making all this happen, without complicated coding and following the verification and validation requirements of the standards.

After all … If it can be written in the manual of operations, it can definitely be programmed in a SIS …

References

1. IEC 61508, Functional Safety of Electrical/Electronic/Programmable Safety-related Systems, Part 1-7, Geneva: International Electrotechnical Commission, 1998.

2. IEC 61511, Functional Safety: Safety Instrumented Systems for the Process Industry Sector, Parts 1-3, Geneva: International Electrotechnical Commission, 2003.

3. ANSI/ISA S84.00.01-2004, Application of Safety Instrumented Systems for the Process Industries, The International Society of Automation, Research Triangle Park, NC, 2004.

4. Goble, W. M., Evaluating Systems Safety and Reliability - Techniques and Applications, NC: Raleigh, ISA 1997.

5. Functional Safety Engineering I & II – Exida LLC 2001 – 2004

6. Goble, W. M., Control Systems Safety Evaluation & Reliability, Research Triangle Park, NC - ISA 1998

7. Simatic Safety Matrix V 6.1 Help Manual, Copy rights 1995-2004, Siemens AG

8. Simatic PCS 7 User's Manual