The safe decision.

White Paper
Functional Safety: A Practical Approach for End-Users and System Integrators

Date: 01 June 2006

Author(s): Tino Vande Capelle
Dr. M.J.M. Houtermans

HIMA Paul Hildebrandt GmbH Co KG
Albert-Bassermann-Strasse 28
68782 Bruehl
Germany
www.hima.com

The safe decision.

Title: **Functional Safety: A Practical Approach for End-Users and System Integrators**

Date: 01 June 2006

Author(s): Tino Vande Capelle
Dr. M.J.M. Houtermans

FUNCTIONAL SAFETY: A PRACTICAL APPROACH FOR END-USERS AND SYSTEM INTEGRATORS

TINO VANDE CAPELLE[1], Dr. MICHEL HOUTERMANS[2, 3]
1- HIMA Paul Hildebrandt GmbH + Co KG, Brühl, GERMANY
2 – Risknowlogy, Brunssum, THE NETHERLANDS
3 - TUV Rheinland Group, Köln, GERMANY
tinovdc@hima.com, m.j.m.houtermans@risknowlogy.com,
michel.houtermans@de.tuv.com http://www.hima.com, http://www.risknowlogy.com

Abstract: - The objective of this paper is to demonstrate through a practical example how an end-user should deal with functional safety while designing a safety instrumented function and implementing it in a safety instrumented system. The paper starts with explaining the problems that exist inherently in safety systems. After understanding the problems the paper takes the reader from the verbal description of a safety function through the design of the architecture, the process for the selection of safety components, and the role of reliability analysis. After reading this paper the end-users understands the practical process for implementing the design of safety instrumented systems without going into detail of the requirements of the standard.

Key-Words: - Functional Safety – Hazard – Risk – Safety Instrumented Systems – Safety Integrity Level –Reliability – PFD – PFH.

1. Introduction

Every day end-users around the world are struggling with the design, implementation, operation, maintenance and repair of safety instrumented systems. The required functionality and safety integrity of safety instrumented systems is in practice determined through a process hazard analysis. This task is typically performed by the end-users as they are the experts on their own production processes and understand the hazards of their processes best. The result of such a process hazard analysis is among others a verbal description of each required safety function needed to protect the process. These safety functions are allocated to one or more safety systems, which can be of different kinds of technology. Those safety systems that are based on electronic or programmable electronic technology need as a minimum to comply with the functional safety standards IEC 61508 and/or 61511.

The end-user is typically not involved in the actual design of the safety system. Normally this is outsourced to a system integrator. The system integrator determines the design of the safety system architecture and selects the safety components based on the specification of the end-users. No matter who designs the safety system according to the safety function requirements, in the end it is the end-user who is responsible for the safety integrity of the safety system. This means that the end-user needs to assure him that the chosen safety architecture and the selected safety components meet the requirements of the applicable standards and be able to defend its decision to any third party performing the functional safety assessment.

In reality end-users and system integrators are not experts in hardware and software design of programmable electronic systems. They know how to run and automate chemical or oil & gas plants but most likely they are not experts on how the operating system of a logic solvers works or whether the communication ASIC of the transmitter is capable of fault free safe communication. Even if they would be experts, the suppliers of the safety components will not give them sufficient information about the internals of the devices so that they can assure themselves of the safety integrity of these devices. Yet they are responsible for the overall design and thus they need to assure themselves that functional safety is achieved. But how can they deal with that in practice?

The objective of this paper is to demonstrate through a practical example how an end-user and/or system integrator should deal with functional safety while designing a safety instrumented function and implementing it in a safety instrumented system. The paper starts with explaining the problems that exist inherently in safety systems. After understanding the problems the paper takes the reader from the verbal description of a safety function through the design of the architecture, the process for the selection of safety components, and the role of reliability analysis. After reading this paper the end-users understands the practical process for implementing the design of safety instrumented systems without going into detail of the requirements of the standard.

2. Why Safety Systems Fail

The hardware of a safety instrumented system can consist of sensors, logic solvers, actuators and peripheral devices. With a programmable logic solver there is also application software that needs to be designed. An end-user in the process industry uses as basis for the design and selection of the safety devices the IEC 61511 standard. This standard outlines requirements for the hardware and software and refers to the IEC 61508 standard if the requirements of the IEC 61511 cannot be not met. This means that even if the IEC 61511 standard is used as basis some of the hardware and software needs to comply with IEC 61508.

As with any piece of equipment also safety equipment can fail. One of the main objectives of the IEC 61508 standard is to design a “safe” safety system. A “safe” safety system means a system that is designed in a way that it can either tolerate internal failures, and still execute the safety function, or if it cannot carry out the safety function any more it at least can notify an operator via an alarm. If we want to design a safe safety system we should first understand how safety systems can fail. According to IEC 61508 equipment can fail because of three types of failures, i.e.,

- Random hardware failures,
- Common cause failures and
- Systematic failures.

2.1 Random Hardware Failures.

Random hardware failures are failures that can occur at any given point in time because of internal degradation mechanisms in the hardware. A typical example is wear out. Any rotating or moving equipment will eventually wear out and fail. There are two kinds of random hardware failures (Rouvroye et. al., 1997):

- Permanent
- Dynamic

Permanent random hardware failures exist until they are repaired. This in contrast to the dynamic random hardware failures. They only appear under certain conditions (for example when the temperature is above 80 C). When the condition is removed the failure disappears again. It is very difficult to test hardware for random dynamic hardware failures.

The IEC 61508 standard addresses random failures in two ways. First of all IEC 61508 requires a designer to implement measures to control failures. The appendix of IEC 61508 part 2 contains tables (Table A16-A18) which represent per SIL level measures that need to be implemented in order to control failures that might occur in hardware.

Secondly, IEC 61508 requires a qualitative and quantitative failure analysis on the hardware. Via a failure mode and effect analysis the failure behaviour of the equipment needs to be analysed and documented. For the complete safety function it is necessary to carry out a probabilistic reliability calculation to determine the average probability of failure on demand of the safety function.

2.2 Common Cause Failures

A common cause failure is defined as a failure, which is the result of one or more events, causing coincident failures of two or more separate channels in a multiple channel system, leading to total system failure. Thus a common cause can only occur if the safety function is carried out with hardware more than once (dual, triple, quadruple, etc. redundancy).

Common cause failures are always related to environmental issues like temperature, humidity, vibration, EMC, etc. If the cause is not related to environmental circumstances than it is not a common cause. Typical examples of a common cause could be a failure of a redundant system due to flooding with water or an EMC field. A common cause failure is only related to hardware, and not to software. A software failure is a systematic failure which is addressed in the next paragraph.

The IEC 61508 standard has two ways to address common cause failures. First of all there is one measure defined to control failures defined, i.e., diversity. Diversity still means that we carry out the safety function in a redundant manner but we use different hardware, or a different design principle or even completely different technology to carry out the same safety function. For example if we use a pure mechanical device and a programmable electronic device to carry out the safety function then a common cause failure of the safety function due to an EMC field will never occur. The programmable electronic device might fail due to EMC but the pure mechanical device will never fail due to EMC.

In practice a real common cause is difficult to find because the failures of a multi channel system must per definition of a common cause occur at exactly the same time. The same hardware will always have different strength and thus fail at slightly a different time. A well designed safety system can take advantage of this gap in time and detect one failure before the other failure occurs.

2.3 Systematic Failures

The most important failures to manage in safety system are the systematic failures. A systematic failure is defined as a failure related in a deterministic way to a certain cause, which can only be eliminated by a modification of the design or of the manufacturing process, operational procedures, documentation or other relevant factors. A systematic failure can exist in hardware and software.

Systematic failures are the hardest failures to eliminate in a safety system. One can only eliminate systematic failures if they are found during testing. Testing that either takes place during the development and design of the safety system or testing that takes place when the system exist in the field (so called proof test). The problem that systematic failures only can be found if a specific test is carried out to find that failure. If we do not test for it we do not find it.

The IEC 61508 standard addresses systematic failure in only one way. The standard defines measures to avoid failures for hardware as well as software. These measures are presented in the appendix of part 2 and 3 of IEC 61508 (respectively tables B1-B5 and tables A1-B9) and depend on the required safety integrity. The standard does not take systematic failures into account in the failure analysis. The philosophy behind this is simple. If all the required measures to avoid failures are implemented and correctly carried out then there are no systematic failures (or at least it is negligible for the desired safety integrity) and thus the contribution to the probability of failure is (close to) zero.

2.4 End-user Responsibility

All though the end-user has no control over the actual design and internal testing of safety equipment ultimately they are still responsible when accidents occur due to any of the three types of failures mentioned above. They need to assure themselves that the safety equipment selected by themselves or their system integrators is compliant with either the IEC 61508 or the IEC 61511 standard. In practice though end-users nor system integrators do not have the knowledge to understand what is going inside safety equipment. They will have to rely on third party assessments of this equipment to assure themselves that the equipment is suitable for their safety application. More on this topic is presented in paragraph 4.

3. FROM HAZARD AND RISK ANALYSIS TO SPECIFICATION TO DESIGN

A safety requirement specification of a safety system must at all times be based on the hazard and risk analysis. A good hazard and risk analysis includes the following steps:

- Hazard identification
- Hazard analysis (consequences)
- Risk analysis
- Risk management
  - Tolerable risk
  - Risk reduction through existing protection layers
  - Risk reduction through additional safety layers

Many techniques exist to support hazard identification and analysis. There is not one ultimate technique that can do it all. A serious hazard and risk study is be based on the use of several techniques and methods. Typical hazard identification techniques include:

- Checklists
- What if study
- Failure mode and effect analysis (FMEA)
- Hazard and operability analysis (HAZOP)
- Dynamic flowgraph methodology (DFM)

Hazard analysis techniques include:

- Event tree analysis (ETA)
- Fault tree analysis (FTA)
- Cause consequence analysis

Risk reduction techniques include:

- Event tree analysis (ETA)
- Layer of protection analysis (LOPA, a variation on ETA)

More techniques exist then the ones listed above that can be used to carry out the hazard and risk analysis. It is important to select the right technique for the right kind of analysis and not to limit oneself to one technique.

3.1 Safety Requirement Specification.

The hazard and risk analysis should among others document in a repeatable and traceable way those hazards and hazard events that require protection via an additional safety function. The results from the hazard and risk analysis are used to create the safety requirement specification of each safety function needed to protect the process. The specification as a minimum defines the following 5 elements for each safety function:

- Sensing
- Logic solving
- Actuating
- Safety integrity in terms of reliability
- Timing

Each safety function description should as a minimum consist of these five elements. The sensing element of the specification describes what needs to be sensed (e.g., temperature, pressure, speed, etc.). The logic solving element describes what needs to be done with the sensing element when it meets certain conditions (e.g., if the temperature goes over 65 C then actuate the shutdown procedure). The actuating element explains what actually needs to be done when the logic solving elements meets the conditions to be met (e.g., open the drain valve).

So far we have described the functionality of the safety function. But the functionality is not complete if we do not know with how much safety integrity this needs to be carried out. The safety integrity determines how reliable the safety function needs to be. The functional safety standards have technical and non-technical safety integrity requirements that are based on the so called safety integrity level. There are four safety integrity levels (1 through 4) where 1 is the lowest integrity level and 4 the highest. In other words it is much more difficult to build a SIL 4 safety function than it is to build a SIL 1 function. The SIL level determines not only the measures to avoid and to control failures that need to be implemented but also the required probability of failure on demand (PFD). The higher the SIL level the lower the probability of failure on demand of this safety function.

The last element to be described is how fast the safety function should be carried out. Also this is a critical element as it depends on the so-called process safety time. This is the time the process needs to develop a potential hazard into a real incident. For example, mixing two chemicals at 30 C is not a problem at all. Mixing the same two chemicals at 50 C can lead to a runaway reaction and result in an explosion. The process safety time is the time the reaction needs to develop into an explosion.

It is common practice in the safety industry to define the time element of the safety function as half of the process safety time. If the chemical reaction takes 2 hours to develop then we have 1 hour to carry out our safety function. On the other hand if the reaction takes 10 seconds we have only 5 seconds to carry out the safety function. It is of up most importance to know this time for two reasons. First of all we need to build a safety function that can actually be carried out in this time. Each device used to carry out the safety function takes a piece of the available total time slot. If we for example use valves that need to be closed we need to make sure that these valves can close fast enough. The second reason is that we need to know whether the build-in diagnostics can diagnose a failure in less than half of the process safety time. Before we need to actuate the safety function we should be able to know whether the safety system has not failed. This puts extra constraints on the internal design of the safety devices when it comes to implementing fast enough diagnostics.

The following is a bad example of a specified safety function:

“The main safety function of the HIPPS is to protect the separation vessels against overpressure and to protect the low pressure equipment against high pressure.”

There is no system integrator who can build the hardware and software from this definition. The only clear aspect is the sensing element. Some where the pressure needs to be measured. After that the system integrator will be lost. The logic, actuating, safety integrity and timing element are not covered with this specification. Specification like this will cost every party involved in the project more time than necessary. It will lead to a lot of unnecessary discussion. A much better example of a safety function specification is the following:

“Measure the pressure on two locations in vessel XYZ and if the pressure exceeds the high-high pressure limit open the drain valve within 3 seconds. Perform the function with a safety integrity of SIL 3.”

This specification gives much more complete information. The system integrator knows exactly what the function should do and can now design the function according to the rules of SIL 3 and select components and write application software that can perform this function.

For each safety function the end-user should provide the system integrator with a clear definition containing as a minimum the 5 elements specified before. There are many other requirements that the end-user can put into the specification. For example environmental conditions that the safety system should be able to handle (temperature ranges, humidity levels, vibration levels, EMC levels, etc.) or restart procedures, periodic test intervals, and more.

A good system integrator will take the safety requirements specification of the end-user and translate that into a requirement specification that is usable for the system integrator. The specification created by the system integrator should be verified and approved by the end-user. This is an excellent step to be performed as it assures that both parties can see that they understand each other and that they interpreted the system to be designed correctly.

Needless to say this costs a more time during specification which is saved during actual design and testing and the often required modifications after words.

3.2 Architectural Design Safety Function

When the safety requirements specification is clear and agreed upon the system integrator can start with the architectural design of the safety function and system. Figure 1 shows how a safety function definition can be implemented in hardware. The safety function is divided into three subsystems, i.e., sensing, logic solving, and actuating. The designer of the safety function can decide how to divide the safety function into subsystem and to what level or detail. In practice subsystems are determined by redundancy aspects or whether the component can still be repaired or not by the end-user.

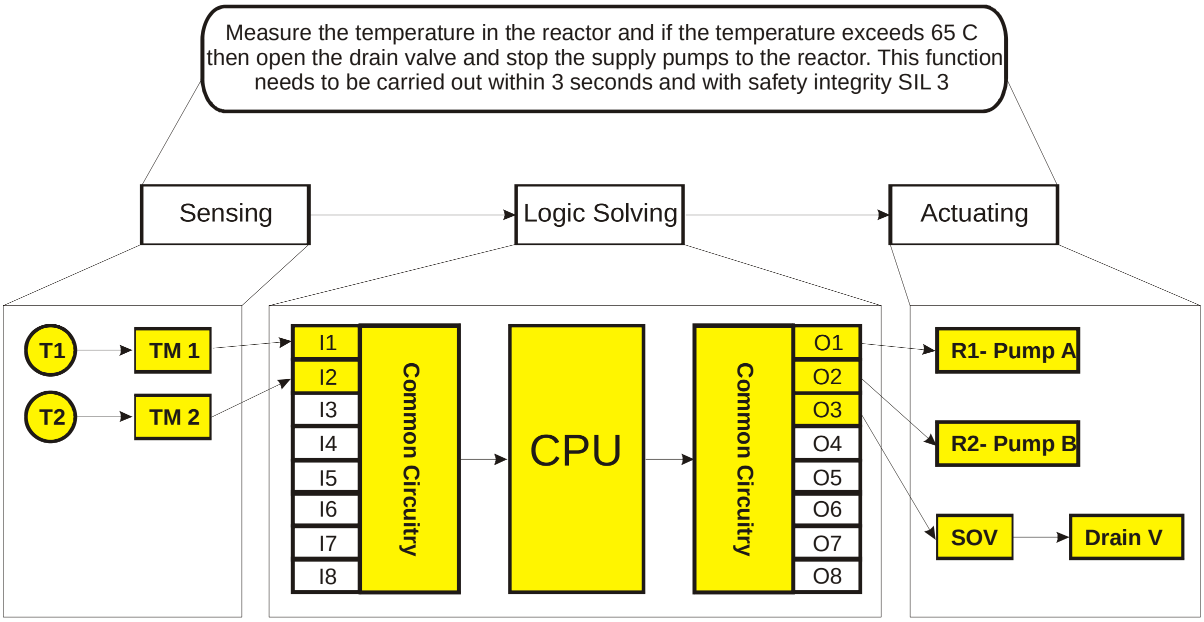

Fig. 1. From specification to hardware design of the safety instrumented system

The IEC 61508 and IEC 61511 standard have set limitations on the architecture of the hardware. The concepts of the architectural constraints are the same for both standards although the IEC 61508 standard requires some more detail. The architectural constraints of the IEC 61508 standard are shown in Table 1 and 2 and are based on the following aspects per subsystem:

- SIL level safety function
- Type A or B
- Hardware fault tolerance
- Safe failure fraction

Table 1 Architectural Constraints Type A

Safe Failure Fraction (SFF)	Type A Subsystem Hardware Fault Tolerance (HFT)		
	0	1	2
< 60 %	SIL 1	SIL 2	SIL 3
60 % -< 90%	SIL 2	SIL 3	SIL 4
90 % -< 99%	SIL 3	SIL 4	SIL 4
> 99 %	SIL 3	SIL 4	SIL 4

Table 2 Architectural Constraints Type B

Safe Failure Fraction (SFF)	Type B Subsystem Hardware Fault Tolerance (HFT)		
	0	1	2
< 60 %	N.A.	SIL 1	SIL 2
60 % -< 90%	SIL 1	SIL 2	SIL 3
90 % -< 99%	SIL 2	SIL 3	SIL 4
> 99 %	SIL 3	SIL 4	SIL 4

The “type” designation of a subsystem refers to the internal complexity of the subsystem. A type A subsystem has a defined failure behaviour and the effect of every failure mode of the subsystem is clearly defined and well understood. Typical type A components are valves and actuators. A subsystem is of type B if only one failure mode and its effect cannot be understood. In practice any subsystem with an integrated circuit (IC) is per definition a type B. Typical type B systems are programmable devices like logic solvers, smart transmitter, or valve positioners.

The hardware fault tolerance (HFT) determines the number of faults that can be tolerated before the safety function is lost. It is thus a measure of redundancy. When determining the hardware fault tolerance one should also take into account the voting aspects of the subsystem. A 1oo3 and 2oo3 subsystem carry out the safety function 3 times (triple redundant) but because of the voting aspect the HFT of the 1oo3 subsystem equals 2 and the HFT of the 2oo3 subsystem equals 1. A complete overview of the most common architectures is given in Table 3.

Table 3 Redundancy versus HFT

Architecture / Voting	Redundancy	HFT
1oo1	No redundancy	0
1oo2	Dual	1
2oo2	No redundancy	0
1oo3	Triple	2
2oo3	Triple	1
2oo4	Quadruple	2

Another important factor is the safe failure fraction (SFF). This is basically a measure of the fail safe design and build-in diagnostics of the subsystem. A subsystem can fail safe or dangerous. Safe failures are those failures that case the subsystem to carry out the safety function without a demand. For example, the safety function of an emergency shutdown valve is to close upon demand. We call it a safe failure if the valve closes because of an internal failure without an demand. A dangerous failure is the opposite. The valve has failed dangerous if it cannot close upon demand because of an internal failure. Some components also have internal diagnostics (diagnostics should not be confused with proof testing). If that is the case it is possible to detect failures and act upon the detection. Smart sensor and logic solvers typically can have build-in diagnostics. Taking this into account a subsystem can basically have four different kind of failures:

- Safe detected (SD)
- Safe undetected (SU)
- Dangerous detected (DD)
- Dangerous undetected (DU)

If we know the failure rates for each subsystem in terms of these four failure categories then we can calculate the SFF as follows:

$$SFF = \frac{\lambda_{SD} + \lambda_{SU} + \lambda_{DD}}{\lambda_{SD} + \lambda_{SU} + \lambda_{DD} + \lambda_{DU}}$$

From the above formula you can see that the SFF is fully determined by the failure rate of the dangerous undetected failures. In other words if we make a fail safe design (lots of safe failures) and we diagnose a lot of dangerous failures (DD) then we will have little dangerous undetected failures and thus a high SFF.

It is important to understand these concepts in order to be able to interpret Table 1 and 2. A system integrator receives from an end-user only the safety function definition with a SIL level attached to it. From the SIL level the system integrator then needs to determine the Type, HFT, and SFF of the subsystem. For example if the system integrator needs to measure the temperature with a subsystem of SIL 3 then there are among others the following options (see Table 1 and 2):

- 1 type A sensor with a SFF > 90%
- 2 type A sensors, 1oo2 or 2oo3, with a SFF 60-90%
- 3 type A sensors, 1oo3, with no diagnostics
- 1 type B sensor with a SFF > 99%
- 2 type B sensors, 1oo2 or 2oo3, with a SFF 90-99%
- 3 type B sensors, 1oo3 with a SFF 60-90%

In other words the system integrator has a lot of design options to choose from. The actual design depends on many things. For example what kind of sensors are available on the market? Which type are they, which SFF do they achieve. Does the end-user have a preferred vendor list to choose from? And so on.

Also the IEC 61511 standard has architectural constraints defined. The principle is the similar as above only the IEC 61511 is less complicated. The IEC 61511 does not differentiate between type A and B components but only between programmable electronic logic solvers and all equipment except programmable electronic logic solvers. Smart sensors with dual processors and software inside are apparently not considered complex devices in terms of IEC 61511. The architectural constraints of IEC 61511 are shown in Table 4 and 5.

Table 4 Architectural Constraints PE Logic Solver

	Minimum hardware fault tolerance (HFT)		
SIL	SFF < 60%	SFF 60% to 90%	SFF> 90%
1	1	0	0
2	2	1	0
3	3	2	1
4	Special requirements apply, see IEC 61508		

Table 5 Architectural Constraints All Equipment Except PE Logic Solvers

SIL	Minimum hardware fault tolerance
1	0
2	1
3	2
4	Special requirements apply, see IEC 61508

For all equipment except PE logic solvers it is possible to decrease the hardware fault tolerance by 1 if the following conditions are met:

- The hardware is prove in use
- Only process related parameters can be adjusted
- Adjustment of process parameters is protected
- The SIL level of the safety function is less than 4

In reality every single product supplier will try to prove to an end-user that their equipment meets the above conditions but in practice it is hard to find a product that truly fulfils these conditions. Specially the proven in use condition is hard to meet, at least the prove for it.

On the other hand the HFT of a product needs to be increased by one if the dominant failure mode of the product is not to the safe mode and dangerous failures are not detected.

4. SELECTING SUITABLE EQUIPMENT

The tables of IEC 61511 and IEC 61508 determine the hardware architecture of the safety function. The starting point is always the SIL level of the safety function and from there the system integrator has a certain degree of freedom to design a safety system architecture depending on the hardware fault tolerance and the hardware complexity of the subsystem.

4.1 Selecting Hardware According to IEC 61511

Having two standards to deal with in order to determine the system architecture does not make it easier for the end-user or system integrator. Many end-users and system integrators do not realize that even if they deal with the IEC 61511 standard that some subsystems of the safety functions still need to comply with the IEC 61508 standard. Figure 2 gives guidance. From this figure it becomes clear that we need per definition to follow IEC 61508 if we want to apply new hardware, which has not been developed yet. For any hardware which meets the IEC 61511 requirements for proven in use or has been assessed according to the requirements of IEC 61508 we can continue to follow the IEC 61511 requirements and particular Table 3 and 4. IEC 61511 defines proven in use as follows:

“When a documented assessment has shown that there is appropriate evidence, based on the previous use of the component, that the component is suitable for use in a safety instrumented system”

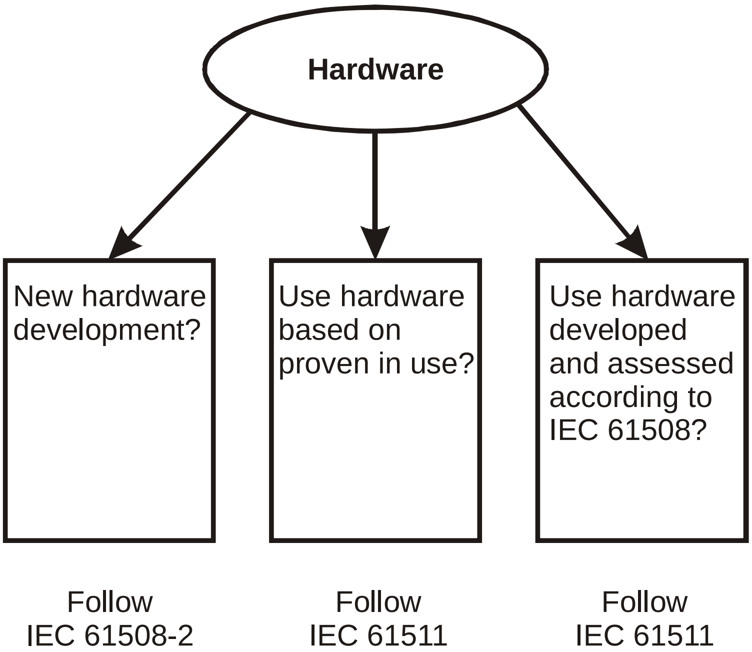

Fig. 2. Which Standard to Follow: IEC 61508 or IEC 61511?

Although proven in use is typically something that only an end-user can determine the suppliers of safety components will do everything to convince end-users and system integrators that their products are proven in use. The evidence though that needs to be delivered in order to “prove” proven in use is not so easy to accumulate:

- Manufacturers quality, management and configuration management systems
- Restricted functionality
- Identification and specification of the components or subsystems
- Performance of the components or subsystems in similar operating profiles and physical environments
- The volume of the operating experience
- Statistical evidence that the claimed failure rate is sufficiently low

Especially the last point is very difficult to meet as failure track records are usually not available. End-user don't always track them and product manufactures do not have the capability to track their products once they are sold and delivered.

4.2 Certification and Third Party Reports

End-users do not have the capabilities to verify for every single product that will be used in a safety function whether it meets the proven in use requirements of IEC 61511 or to assess them according to IEC 61508. Many end-users therefore make use of certified products or third party reports. There is a big difference between a product with a certificate and a product with a third party report.

When a product is certified according to the IEC 61508 standard then this means that every single requirements of the standard is verified for this product. It is for example not possible to only certify the hardware of a programmable electronic system. Certification is all-inclusive and thus also the software needs to be addressed. A well certified safety product not only addresses functional safety according to IEC 61508 but also issues like:

- Electrical safety
- Environmental safety
- EMC/EMI
- User documentation
- Reliability analysis

A certified product always comes with a certificate and a report to the certificate. The report to the certificate is very important as it explains how the verification or assessment has been carried out and whether there are any restrictions on the use of the product.

A third party report is often used in industry but is only limited in scope. The report itself will outline what the scope of the analysis is. Many third party reports only focus on the hardware analysis of the product. In principle this is no problem as long as the end-user or system integrator is aware that other aspects of the product, like the software, also need to be addressed and may be another third party report should be requested that covers the software.

4.3 Required hardware functional safety information

For each safety device the end-user should assure themselves that the device is either compliant with IEC 61508 or with IEC 61511. Concerning the hardware the end-user should as a minimum ask from their suppliers the information listed in Table 6.

Table 6 Hardware Checklist

Item
Applicable standard
Type
Hardware fault tolerance
Safe failure fraction
Safe detected failure rate
Safe undetected failure rate
Dangerous detected failure rate
Dangerous undetected failure rate
SIL level for which the product is fit for use
Recommends periodic proof test interval

With this information the end-user or system integrator can easily determine how to comply with the architectural constraints tables and build the architecture of their loop as desired. This information can be delivered by the supplier itself, through a third party report or through a certification report. It is up to the end-user to decide what is actually required (read what can be trusted). The architecture needs to be redesigned until the architectural constraints requirements are met.

5. THE ROLE OF RELIABILITY ANALYSIS

Once the architectural system design of the safety loop complies with the architectural constraints tables the loop has met already one of the most important requirements of the standards. Another important requirement is the probability of failure on demand (PFD) or continuous mode (PFH) calculation. This is the probability that the safety function cannot be carried out upon demand from the process. It needs to be calculated for those processes where the expected demand is less than once per year. If a loop is used in a continuous mode then it is necessary to calculate the frequency of failure of this loop per hour. This is necessary as we now are in a different situation. Where the demand loop can only cause a problem when there is an actual demand from the process the continuous loop can actual be the cause of a process upset when the loop itself has failed. Table 7 gives an overview of the required probabilities and frequencies per SIL level.

Table 7 PFD versus PFH

	Demand Mode	Continuous Mode
SIL	Probability of failure on demand	Frequency of the failure per hour
4	$\geq 10^{-5}$ to $< 10^{-4}$	$\geq 10^{-9}$ to $< 10^{-8}$
3	$\geq 10^{-4}$ to $< 10^{-3}$	$\geq 10^{-8}$ to $< 10^{-7}$
2	$\geq 10^{-3}$ to $< 10^{-2}$	$\geq 10^{-7}$ to $< 10^{-6}$
1	$\geq 10^{-2}$ to $< 10^{-1}$	$\geq 10^{-6}$ to $< 10^{-5}$

In order to carry out a probability or frequency calculation the following information is required:

- A reliability model per loop
- The reliability data for all equipment in the loop

5.1 Reliability modeling

A reliability model needs to be created for each loop of the safety system. There are different techniques available in the world to create reliability models. Well known techniques include:

- Reliability block diagrams
- Fault tree analysis
- Markov analysis

The reliability block diagram technique is probably one of the simplest methods available. A block diagram is a graphical representation of the required functionality. The block diagram of the safety function of Figure 1 is given in Figure 3 below. A reliability block diagram is always read from left to right where each block represents a piece of the available success path(s). As long as a block has not failed it is possible to go from left to right. Depending on build-in redundancy it is possible that alternative paths exist in the block diagram to go from left to right. Once the block diagram is created it is possible to use simple probability theory to calculate the probability of failure.

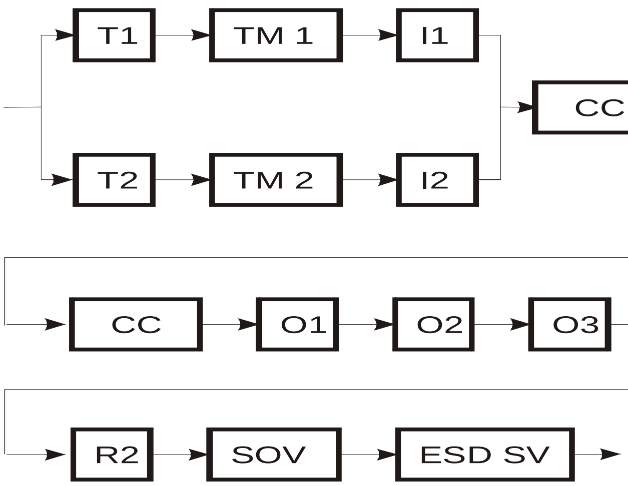

Fig. 3. Block diagram safety function

Another technique is fault tree analysis (FTA). FTA is a technique that originates from the nuclear industry. Although the technique is more suitable to analysis complete facilities it is also used to calculate the probability of failure on demand of safety loops. A FTA is created with a top even in mind, e.g., safety does not actuate on demand. From this top event an investigation is started to determine the root causes. Basically an FTA is a graphical representation of combinations of basic events that can lead to the top event. A simplified version of the FTA for the safety function in Figure 1 is given in Figure 4. It is possible to quantify the FTA and calculate the probability of occurrence of the top even when the probabilities of occurrence of the basic events are known.

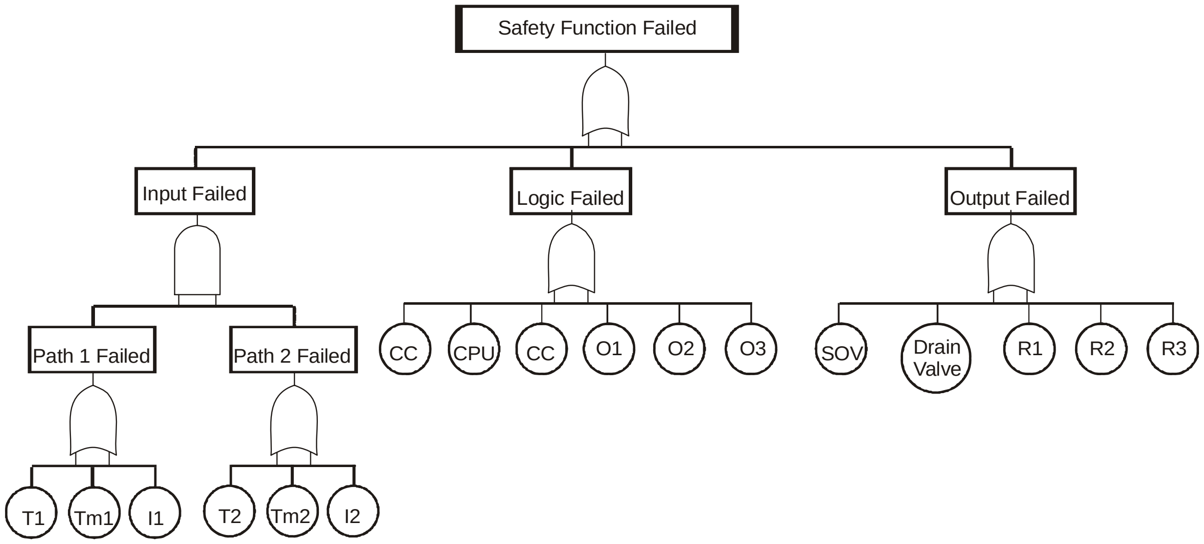

Fig. 4. Simplified fault tree diagram safety function

Research has indicated that Markov analysis is the most complete and suitable technique for safety calculations. Markov is a technique that captures transitions between two unique states. In terms of safety this means the working state and the failed state. Going from one state to the other can either be caused by a failure of a component or by repair of a component. Therefore a Markov model is also called a state transition diagram. See Figure 5 for the Markov model of the safety function of Figure 1. Once the Markov model is created and the rate of transition is known between two states (that is the failure rate or repair rate) it is possible to solve the Markov model and calculate the probability of being in a state.

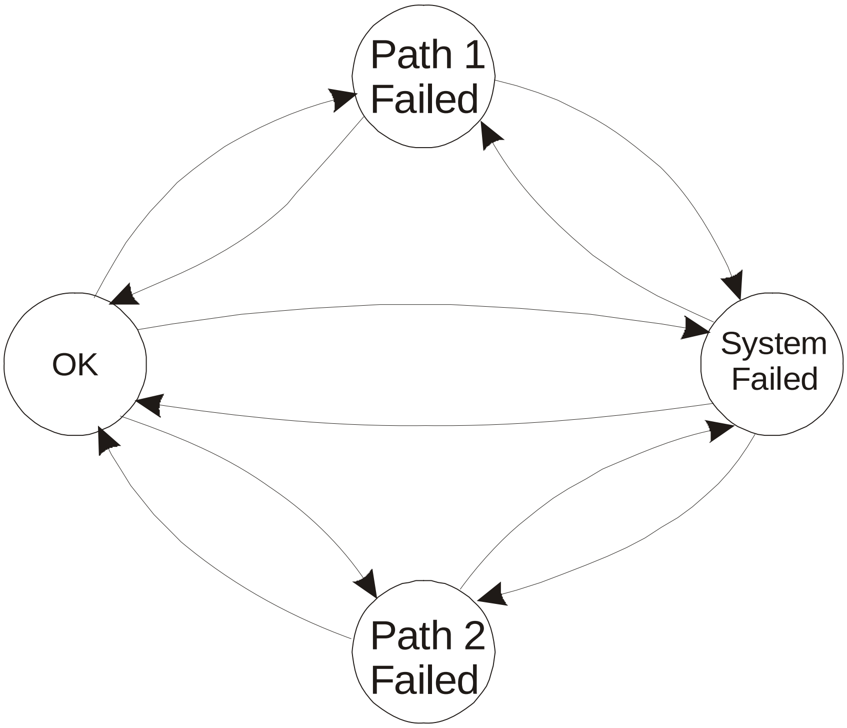

Fig. 5. Markov model safety function

The IEC 61508 standard also has standard formulas to calculate the PFD and PFH for each loop. These formulas are called simplified equations. What many people do not realize though is that these simplified equations are derived from Markov models and that in practice it is not so simple to derive them. Another limitation of these equations is that they only exist for 1oo1, 1oo2, 2oo2, 1oo2D, and 2oo3 architectures. For any other kind of architecture the standards do not provide equations and thus one needs to refer to any of the above mentioned techniques. Also the simplified equations are not flexible enough to handle diverse equipment, or different repair times and periodic proof test intervals. Hence their name simplified. For a complete list of simplified equations derived from Markov models see Börcsök (2004). The following two equations are examples of the simplified equations as they can be found in the standards

1oo1 PFD equations:

$$PFD_{avg} = \left(\lambda_{du} + \lambda_{dd}\right)\cdot t_{CE}$$

1oo2 PFD equations:

$$PFD_{avg} = 2\left(\left(1-\beta_D\right)\lambda_{dd} + \left(1-\beta\right)\lambda_{du}\right)^2 t_{CE} t_{GE} + \beta_D \lambda_{dd} MTTR + \beta\lambda_{du}\left(\frac{T_1}{2} + MTTR\right)$$

Rouvroye (1998) has compared different reliability techniques and their usefulness in the safety industry. Figure 6 gives an overview of these techniques and the result is that Markov analysis is the most complete technique. With Markov it is possible to create one model that allows us to take into account any kind of component, diverse component, different repair and test strategies. It is possible to calculate the probability of failure on demand, the rate per hour or the probability that the safety system causes a spurious trip. No other technique can do this.

	Safety ranking comparison	Availability prediction	Probability of Unsafe Failure	Effects of test & repair	Trip rate prediction	Time dependent effects
Expert analysis	✓	✓				
FMEA	✓					
FTA	✓	✓	✓	?	✓	
Reliability Block Diagram	✓	✓	✓			
Parts Count Analysis	✓	✓ ?	✓			
Markov Analysis	✓	✓	✓	✓	✓	✓

Fig. 6. Comparison reliability techniques for safety analysis (Rouvroye, 1998)

5.2 Reliability data

Every reliability model needs reliability data in order to actually perform the reliability calculation and quantify the results. There are many sources for reliability data. The following list is an overview of available data:

- End user maintenance records
- Databases
- Handbooks
- Manufacturer data
- Functional safety data sheets
- Documented reliability studies
- Published papers
- Expert opinions
- Reliability Standards

The absolute best data an end-user can use is its own maintenance data. Unfortunately not many end-users have their one reliability data collection program and there is of course always the problem that a new safety system contains devices that were not used before by the end-user. Luckily there are more and more databases available were there is a collection of data from different sources which can be used by end-users.

For the calculation we need the following reliability data for each device:

- Safe detected failure rate
- Safe undetected failure rate
- Dangerous detected failure rate
- Dangerous undetected failure rate

This data was already collected when the architectural constraints were verified, see Table 5. On plant level we also need to know the following reliability data:

- Repair rate per device
- Periodic proof test interval
- Common cause

The repair rate per device depends on the availability of the spare device and the availability of a repair crew. Periodic proof test intervals can be determined by three means:

- The supplier of the device specifies a rate
- Laws or standards determine a minimum inspection interval
- The desired SIL level determines the period proof test interval through the PFD calculation

Once the reliability model is created and the reliability data is collected the actual calculation can be performed. Figure 7 shows an example of PFD calculation with and without periodic proof testing. Actually every year an imperfect proof test is performed which assures that the PFD level of the safety function stays within the SIL 3 range.

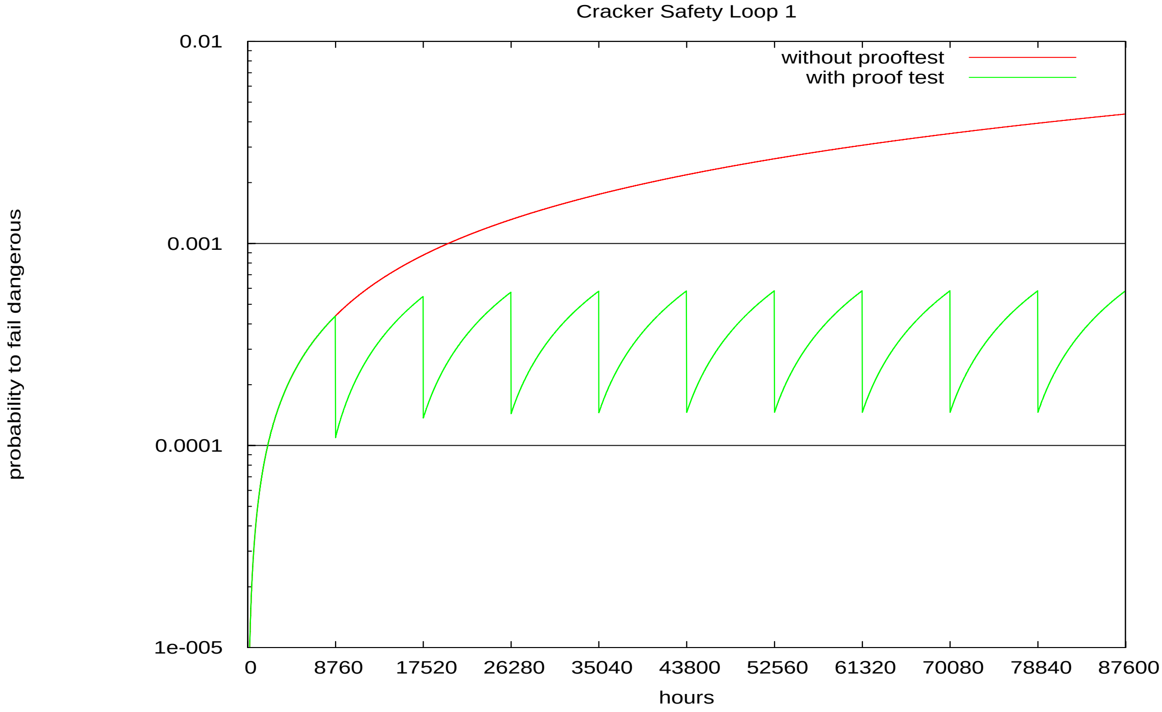

Fig. 7. Example PFD calculation with and without proof testing

6. CONCLUSIONS

This purpose of the paper was to explain to end-users the most important high level requirements when designing safety instrumented system. The paper explained that safety system can fail in three different ways and that it is important to design, operate and maintain a safety system in a why that those three failures types are controlled. In order to understand the functional requirements of a safety system it is important to carry out hazard and risk analysis. The paper explained several techniques that can be used for this purpose. The results of the hazard and risk analysis are documented as the safety function requirement specification. Next the paper explained from a top level safety function description the high level requirements that end-users or system integrators need to follow in order to design the actual safety instrumented systems. The paper explained the significance of the architectural constraints from the point of view of the IEC 61508 and IEC 61511 standard. For end-users and system integrators it is important to collect reliability data and to perform reliability analysis and be able to calculate the safe failure fraction and the probability of a failure on demand calculations.

References:

[1] Rouvroye, J.L., Houtermans, M.J.M., Brombacher, A.C., (1997). Systematic Failures in Safety Systems: Some observations on the ISA-S84 standard. *ISA-TECH 97,*ISA, Research Triangle Park, USA.

[2] IEC (1999). *Functional safety of electrical, electronic, programmable electronic safety-related systems, IEC 61508*. IEC, Geneva.

[3] IEC (2003). *Functional safety – safety instrumented systems for the process industry, IEC 61511.* IEC, Geneva.

[4] Houtermans M.J.M., Velten-Philipp, W., (2005). The Effect of Diagnostics and Periodic Proof Testing on Safety-Related Systems, TUV Symposium, Cleveland, OHIO.

[5] Börcsök, J., Electronic Safety Systems, Hardware Concepts, Models, and Calculations, ISBN 3-7785-2944-7, Heidelberg, Germany, 2004