Commissioning a large machine **functional safety project**

Ian Hetherington, VANTAGE

Methods on how the client's User Requirement is Specified (URS) and the recording of the verification and validation procedure.

This article is taken from the aspect of the client or end user. They may be putting a single machine or a large complex of machines into service, the question remains the same. Did they get the safe system they required? It is not the role of the end user to design and specify out the complete detail of the safety system. It is certainly their role to verify and validate that the system performs to the required level. To this end the article suggests methods on how the client's user requirement is specified (URS) and the recording of the verification and validation procedure.

Systematic and installation errors

To begin with may I put this question to you? From a performance aspect, what is the difference between a regular control system and a Safety Related Control System (SRCS)? A regular control system has an independent and continuous validation of its performance. This is provided by the key performance indicators for the process under control, such as quality and efficiency. The very reason for the control system's existence is being challenged on an hour by hour, day by day basis. Can the same be said of an SRCS? Not really, no it cannot. An SRCS may only be challenged when a demand is placed on it. There is no independent and continuous validation of its quality of performance.

The quality of a regular control system is measured in the very fine confectionary it produces or the excellent motor cars it produces. The scale of quality of an SRCS is measured in two possible ways. Performance Level (PLr) or Safety Integration Level (SIL) If that PLr or SIL was inherently wrong on day one, it will not show up in the quality of that shiny new car or that tasty biscuit. Therefore for the team involved in the delivery of a safety system, it is imperative that a Functional Safety Management Plan is effective in reducing the possibility of systemic design errors and installation errors. There are two regulated or standard approaches this management plan:

- IEC 62061: Functional Safety Programmable Systems
- ISO 13849: Safety Related Parts of a Control System

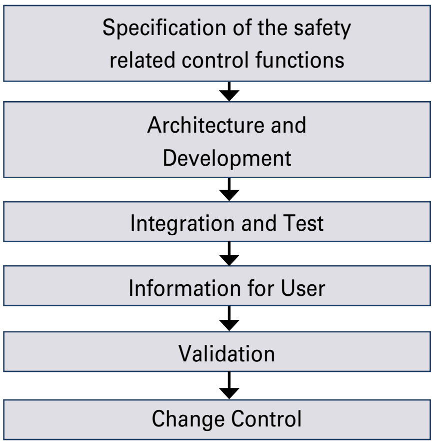

Specification of the safety related control functions
Architecture and Development
Integration and Test
Information for User
Validation
Change Control

Figure 1: Basic Functional Safety Management Plan.

It is not the intention of this article to discuss the detail of functional safety management plans in either standard IEC 62061 or ISO 13849; neither to discuss the management of the design of safety systems. It is rather the intention to discuss the practicable application of a management plan under such topics as:

CRC	– Cyclic Redundancy Check
EDM	– External Device Monitoring
I/O	– Input/Output
IEC	– International Electrotechnical Commission
ISO	– International Standards Organisation
MTTFd	– Mean Time To dangerous Failure
PLC	– Programmable Logic Controller
PLr	– Performance Level
SIL	– Safety Integration Level
SRCS	– Safety-Related Control Systems
URS	– User Requirement Specification

Abbreviations/Acronyms

- Avoiding excessive or cumbersome management plans
- Modular approach to functional design specification (URS)
- Recording the verification process
- Recording the validation process using the URS

Other subsidiary topics for discussion are:

- Detecting and final control Elements
  - The rise of programmable or more accurately 'parameterable' (if that's a word) elements presents its own set of challenges
  - Traditional detecting and final elements had a dedicated single function. It did exactly 'what it said on the tin'
  - These elements continue to develop with self-teach functions, floating muting, profile for safe minimum speed, etc.
  - One must take care that with all the available flexibility, that the required safety function is being executed
- Areas of most frequent sub-standard design

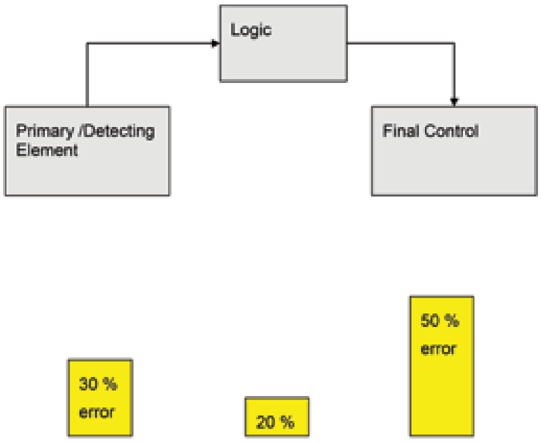


Figure 2: Areas of most frequent sub-standard design.

These figures are not based on recorded statistics, but from observation of many projects. The high error rate for final control elements, is mainly due to non-safe rated components in a safety loop without sufficient diagnostics, redundancy or insufficient Mean Time To dangerous Failure (MTTFd).

Modular User Requirement Specification

On a large or more complex project, where there are multiple suppliers of major sections of plant, in addition these suppliers may be from different countries with varying statuary regulation. This places greater emphasis the URS and the Safety Management Plan. Hypothetically, we are considering a project which covers many hundreds of square metres, several thousand I/O (regular control) and different complexes of machinery. It is a production process using a variety of complex machinery. We are discussing the delivery of the safety system for this. A modular approach to requirement specification of safety functions and then building these modules into safety loops, creates a clear and unambiguous statement. In broad terms, typical safety loops can be grouped into the following (the list is not definitive):

- E-Stop
  - Zoning
  - Class of stop function (break for free run)
- Access Control
  - Physical restraint with interlock or guard locking
- Presence sensing
- Muting or Bypass
  - Safe speed
  - Hold to run / Jog
- Process interlock
  - Hazard materials

Rather than specifying the detail function of each complete safety loop, of which there may be several hundred in a large complex project, one chooses the modules that make up the loop. Some of the benefits to this type of development are:

- Avoids repetition of stating the same function in each safety loop
- Transparent to the hardware or software platform being used
- Diagrammatic format tends towards a clear, and unambiguous definition
- A revision of a module does not require it to be exhaustively revised in every loop. Change it once at the module definition and it is referenced to wherever it is called

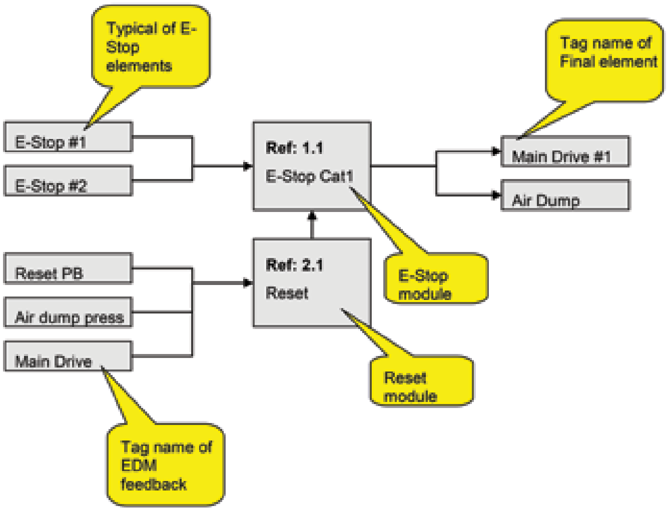


Figure 3: Example E-Stop loop.

In the example the module Ref 1.1 E-Stop and the module Ref: 2.1 Reset are specified for this particular safety loop. The Ref: 1.1 E-Stop may be re used again and again in other E-Stop loops. All that changes are the tag names of the input elements and output elements.

Definition of a module E-Stop Cat1 (Break Stop)

Refer to ISO 13850 E-Stop principles of design.

- E-Stop push buttons shall be dual channel (min Cat 3 architecture)
- There is no zoning of E-Stop functions. The E-Stop shall be global to the defined area
- E-Stop contacts shall be normally closed of the self-monitoring type, see hardware specification
- E-Stops shall adhere to the requirements of ISO-13850:2008
- The E-Stop category shall be Cat 1. i.e. break stop

It is recognised that the function of the E-Stop is to avert arising or reduce existing hazards to persons, damage to machinery or to work in progress. It is not a substitute or alternative to any protective measure such as a safety interlock to prevent access to mechanical movement.

Reset (Manual Monitored)

- All E-Stop functions shall be Monitored Manual Reset, requiring External Device Monitoring (EDM) with the exception of safe rated final elements with self-monitoring
- The reset shall be taken from the falling edge of the reset pulse
- The reset pulse shall be 'AND' with the EDM
- The reset command shall not be accessible from within the hazard area

Other examples might be presence sensing i.e. light curtains. The behaviour of that particular module will define how it will respond to inadvertent access – in other words a shut down to a safe condition. It will also define the behaviour under muting conditions, what sequence it will have and time out, etc.

Recording of the verification process

The objective of the verification by analysis is to establish if the SRCS shall function correctly and if it attains the required safety performance level or SIL. IEC 62061 in particular calls for details about strategy, role and identification of the people involved etc. There are different techniques to adopt. The 'top down' approach such as Fault Tree Analysis, or in the example below, the 'bottom up' approach. At a minimum the following is required to record the process. There are a number of core documents required. *Figure 4* is a flow diagram showing how the documents support the analysis to determine if the safety loop 1. Functioned correctly and 2. Did it achieve the required safety level?

> *If the Performance Level (PLr) or Safety Integration Level (SIL) PLr was inherently wrong on day one, it will not show up in the quality of that shiny new car or that tasty biscuit.*

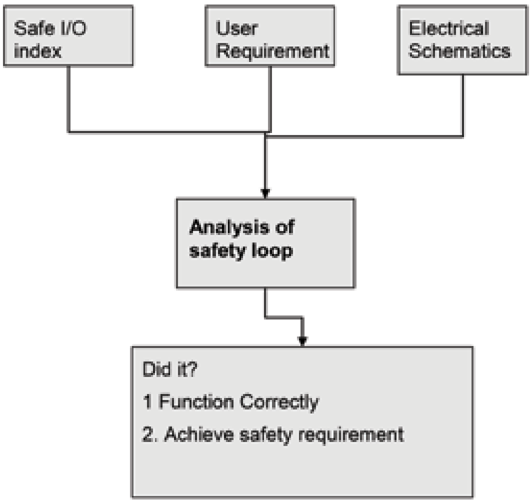


Figure 4: Flow diagram of analysis.

Example of analysis

- From the documentation the inputs and outputs for this E-Stop safety loop are defined
- From the URS the function of the E-Stop and Reset are defined
- Error: From analysis it is found that there is no safe message being passed to the final elements, and the Reset is Auto reset, it should be manual

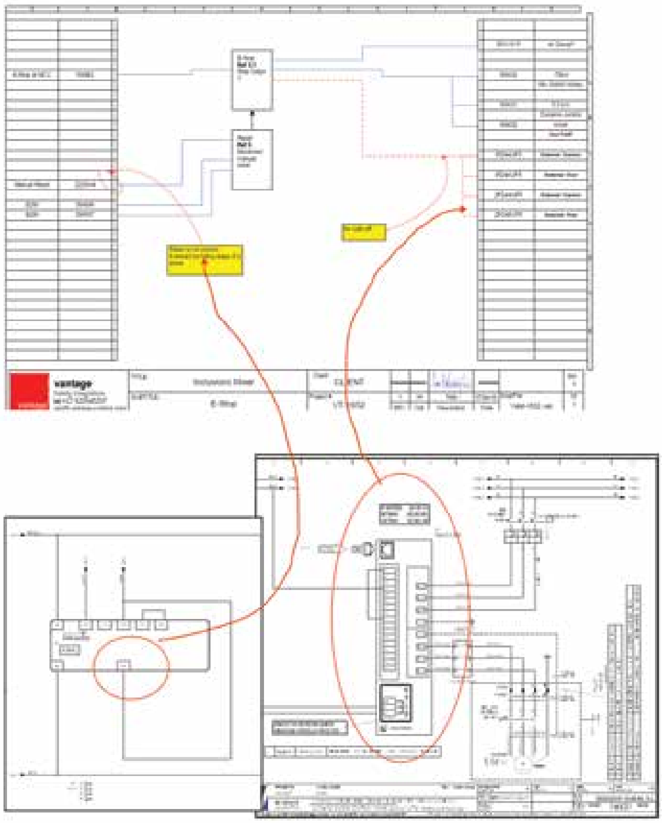


Figure 5: Example of analysis.

By using the graphical method, proof of the analysis is recorded and errors of design are identified. It should be noted that the work previously undertaken in the URS in defining modular functions is expanded further by defining the I/O for a particular loop. The logic of how the safety elements are configured is analysed in the electrical schematic and recorded in this format. In the case of safe software the Tag-Name and PLC address are recorded on diagram.

Recording of the validation process

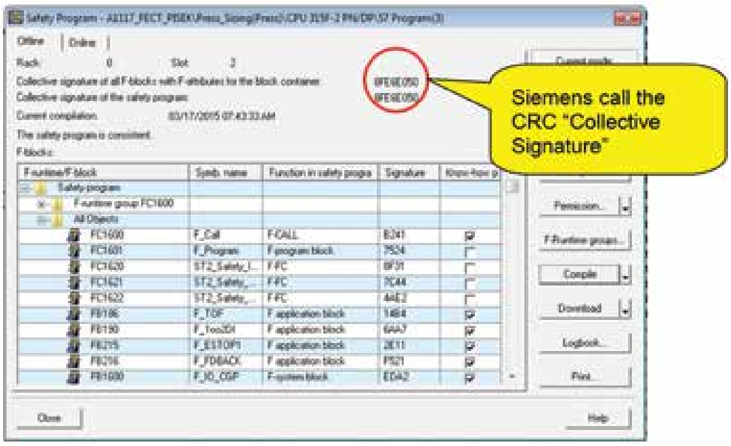
Validation in this case, is taken to mean validation of the installed safety system. Testing of each installed detecting element trough the logic to each resultant final control element's behaviour. The validation test may only be satisfactorily undertaken after the analysis of the safety loops that make up the safety system. In other words, one has to understand the safety function to be able to test it. For example: A fault in how a final element is safely shut down may be masked by the regular control system stopping that element. Therefore the test must observe that the safety message is being passed to the final element. For large complex systems, simulation of the safe logic is useful in debugging, I do not see it as a substitute for a validation test. A 'Shut Off Matrix' is simple and somewhat useful when the safety loop is basic. Where it is found lacking is in the reset function after safe shut off. It does not address the reset action or device coverage. Cause and Effect test sheets are also useful, but can become exhaustive and cumbersome.

Use if modular functions for validation

Previously in the URS, the behaviour of various modules making up a safety loop where defined Carrying those URS modules forward to the validation phase, they are used as a model or template against which the action of the loop under test is witnessed. The function block type test shown is a suggested method for recording validation tests. The modules defined in the user specification are carried forward to the validation test. Where a safety loop is more complex, with various parameters to enable different safe operating conditions. For example a hand held jog station or enabling pendant. There may be a number of test sheets for the one loop. The test sheet begins with all inputs and out puts marked in red. As the test progresses with positive results the input and outputs are marked in green. When the complete loop is tested satisfactorily it is signed by the responsible person. Observations and notes may be added in yellow.

Safe software reporting

Most safe software platforms provide a Cyclic Redundancy Check (CRC). This is shown as hexadecimal number. Once the final software configuration is complete and compiled it will generate this CRC number. This number is unique to the configuration at that time, any further changes will generate a different CRC number Therefore the recording of the validation test must include the CRC number.



Safety Program - A1117_FECT_PISEK\Press_Sizing\Press\CPU 315F-2 PN/DP\S7 Program(3)

Offline | Online

Rack: 0 Slot: 2

Collective signature of all F-blocks with F-attributes for the block container: 8FE6E050

Collective signature of the safety program: 8FE6E050

Current compilation: 03/17/2015 07:43:33 AM

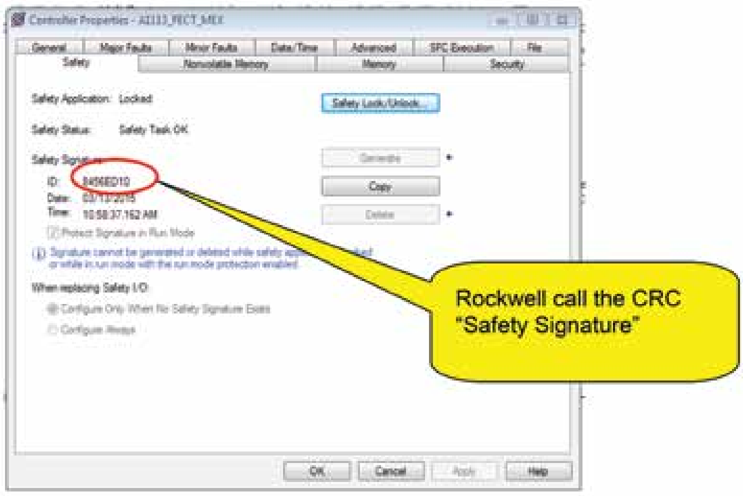
The safety program is consistent.

F-blocks:

F-runtime/F block	Symb. name	Function in safety progra	Signature	Know-how p
Safety program				
F-runtime group FC1600				
All Objects				
FC1600	F_Call	F-CALL	B241	☑
FC1601	F_Program	F-program block	7524	□
FC1620	ST2_Safety_I...	F-FC	8F31	□
FC1621	ST2_Safety_...	F-FC	7C44	□
FC1622	ST2_Safety_...	F-FC	4AE2	□
FB186	F_TOF	F-application block	14B4	☑
FB190	F_Two2DI	F-application block	6AA7	☑
FB215	F_ESTOP1	F-application block	2E11	☑
FB216	F_FDBACK	F-application block	F521	☑
FB1600	F_IO_CGP	F-system block	EDA2	☑

Siemens call the CRC "Collective Signature"

Figure 6: Example Siemens Step7-F. CRC (Siemens calls the CRC 'Collective Signature').



Controller Properties - A1113_FECT_MEX

Safety Application: Locked

Safety Status: Safety Task OK

Safety Signature

ID: 8456ED10

Date: 03/13/2015

Time: 10:58:37.162 AM

☑ Protect Signature in Run Mode

Signature cannot be generated or deleted while safety application is locked or while in run mode with the run mode protection enabled.

When replacing Safety I/O:

● Configure Only When No Safety Signature Exists

○ Configure Always

Rockwell call the CRC "Safety Signature"

Figure 7: Example Rockwell GuardLogix.CRC (Rockwell calls the CRC 'Safety Signature').

Ian Hetherington is a certified machine safety engineer. He provides design, specification and CE certification of safety systems. Also independent analysis and validation of safety systems. Technical files are auditable and verifiable to ISO/IEC and SANS standards. Projects are undertaken in South Africa, Europe and United States.

Enquiries: Email ian@vantage-control.com