PRQA Programming Research

WHITE PAPER

Achieving machinery functional safety according to IEC 61508, ISO 13849 and IEC 62061

Silvano Sogus
November 2015

This paper discusses functional safety in machinery, according to the relevant industry standards such as IEC 61508, ISO 13849 and IEC 62061

Unlike standards such as ISO 26262 (automotive) or EN 50128 (railways), which stand clearly as well-accepted references for specific applications, machinery projects can refer to two distinct type B guidelines – ISO 13849 and IEC 62061. These are both derived from the parent standard IEC 61508, which stands as reference for the parts ISO 13849 and IEC 62061 do not cover. We will highlight how the standards cope with embedded software development; we will then provide a description of PRQA's tools and how these can be deployed to help comply with functional safety in machinery applications

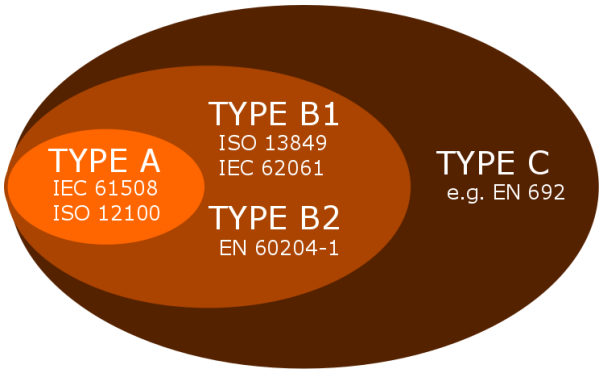
Introduction

Machinery is a market that benefits from the increasing introduction of automation-enabled efficiency. Whether taking into consideration simple activities involving one operator – such as a saw workstation – or more obviously complex structures - such as a robot welding systems - the integration of advanced features and the increase in computational power has been steady and consistent. As PLCs have traditionally dominated the market – due to their "hard" real-time properties – new embedded general-purpose platforms are working their way into common machinery adoption. [1]

However, standardization processes have not kept the same pace; today, the most widespread industry standards – ISO 13849 and IEC 62061 – address only part of the problem of dealing with software in machinery, and almost exclusively at application level. Users, producers and system integrators have to revert to the general indications in IEC 61508-3.

Standards in machinery

Standards for machinery applications can be classified according to a three-level hierarchy: type A (IEC 61508, ISO 12100) provide basic design guidelines and basic terminology for machinery; type B are divided into type B1 (ISO 13849, IEC 62061), which cover general safety aspects, and type B2 (EN 60204-1) that provide reference to special protective devices. Finally, type C (for example EN 692) defines specific features for individual machinery groups. Base levels are more authoritative than top levels; meaning type C standards can overrule type B and type A.



ISO 13849-1:2006 "*Safety of machinery -- Safety-related parts of control systems -- Part 1: General principles for design*" provides safety requirements and guidance on the principles for the design and integration of safety-related parts of control systems (SRP/CS), including the design of software. From 1st January 2012 it superseded EN 954-1 in providing presumption of conformity to the new European Directive 2006/42/EC on machinery. It is technology agnostic in that it can be applied to electrical, hydraulic, pneumatic, mechanical systems and so on.

IEC 62061 "*Safety of machinery: Functional safety of electrical, electronic and programmable electronic control systems*" is the machinery specific implementation of IEC 61508. It provides requirements that are applicable to the system level design of all types of safety-related electrical control systems and for the design of non-complex subsystems or devices. It requires that complex or programmable subsystems should satisfy IEC 61508.

A merging process that will join ISO 13849-1 and IEC 62061 into a single standard (ISO/IEC 17305), although already in place, seems to be suffering from difficulties and slow adoption.

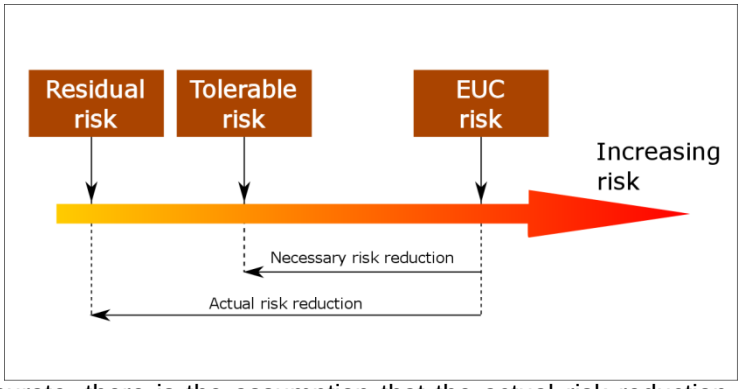
Scope of applicability

A first level fundamental distinction between these two standards is the scope of application. IEC 62061 is limited to electrical systems, while ISO 13849-1 can also be applied to other technologies, such as pneumatic and hydraulic mechanical systems.

As a guideline, the following decision flow could be adopted to determine which of the two standards should be used depending on the specific conditions of the application: [2]

- IEC 62061 is suggested if there are uncommon safety functions to be implemented, or if the system requires complex programmable electronics to a high level of integrity; the methodology of IEC 62061 has been designed to work with complex safety function that might require sophisticated system architectures.
- ISO 13849-1 is for common, conventional safety function implementations, or for when the target is not an electrical system.

Performance Levels and Safety Integrity Levels



Another important difference between ISO 13849 and IEC 62061 is in the way the risk reduction level associated to a specific safety function is defined and calculated.

The necessary risk reduction is the amount of reduction required to lower the risk level from the current value (EUC risk (equipment under control)) and the risk level deemed to be tolerable; this is achieved by safety functions, based on the characterization of the single risk. Since risk estimates are by definition inaccurate, there is the assumption that the actual risk reduction achieved is different (higher) than the necessary one, therefore the residual risk is not equal to the tolerable risk, but lower. Among the risk coverage technologies, the main focus of IEC 61508 is on E/E/PE (electrical, electronic and programmable electronic) systems.

According to ISO 13849, SRP/CSs (Safety-Related Part of a Control System) , in order to enable the system to operate safely, are required to have capabilities simplified according to five PL (performance levels) a, b, c, d and e. There is some correspondence between PL and SIL (software integrity level) as defined in IEC 61508/ IEC 62061. The highest level considered in machinery is SIL 3/PLe. IEC 62061 does not consider SIL 4 risk reduction level as the projected harm in the machinery sector is rarely more than one fatality, whereas IEC 61508 applies to sectors that can credibly result in multiple fatalities. [3]

ISO 13849 Performance Levels	Risk reduction	IEC 61508/IEC 62061 Safety Integrity Levels
PLa		-
PLb		SIL1
PLc		
PLd		SIL2
PLe		SIL3

The level of risk reduction that a particular safety function should achieve sets the PLr (required PL) or the SIL for that safety function. Examples of safety functions can be:
- pushing the local emergency-stop button so the motor mode goes to STO (safe torque off). This ensures that no torque-generating energy is supplied and that non-intentional start-up is prevented.
- if people and/or objects are inside a specific area, motor mode changes from SLS (safely-limited speed, reduce the speed under a safe level) to STO.

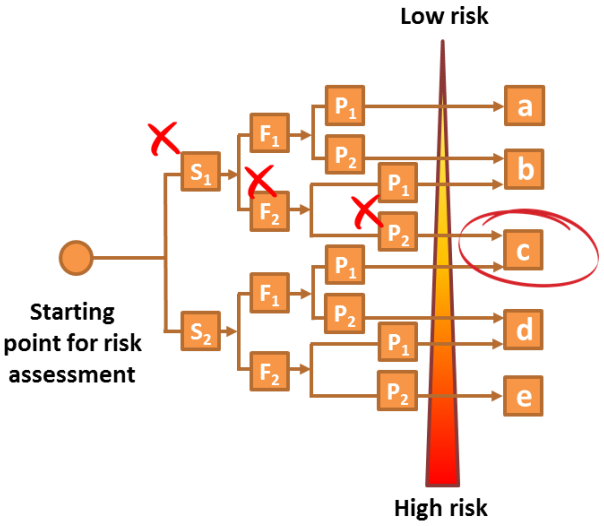
Risk reduction assessment

The way to determine the required risk reduction level to be achieved for each safety function is different in IEC 62061 and ISO 13849-1.

In ISO 13849-1, the PLr is identified as a result of quantification of severity of injury (S), frequency and/or exposure to hazard (F) and possibility to avoid the hazard or limit harm (P).

IEC 62061 introduces the probability of the hazardous event (W). F, W and P are added together to define the class Cl of the resulting safety integrity level. The final SIL is therefore found to cross the class range where Cl falls with the severity of the harm. The AM result is a recommendation to apply other measures.

EN ISO 13849-1



Severity of injury (S)	Frequency and/or duration of exposure (F)	Possibility to avoid the hazard (P)
S_1 – Slight (normal reversible injury)	F_1 – Seldom to quite often and/or the exposure time is short	P_1 – Possible under specific conditions
S_2 – Serious (normally irreversible injury including death)	F_2 – Frequent to continuous and/or the exposure time is long	P_2 – Scarcely possible

IEC 62061

Frequency and duration	F > 10 min	F ≤ 10 min	Probability of hazardous event	W	Avoidance	P
≤ 1 hour	5	5	High	5		
> 1 hour - ≤ 1 day	5	4	Likely	4		
> 1 day- ≤ 2 weeks	4	3	Possible	3	Impossible	5
> 2 weeks- ≤ 1 year	3	2	Rarely	2	Possible	3
> 1 year	2	1	Negligible	1	Likely	1

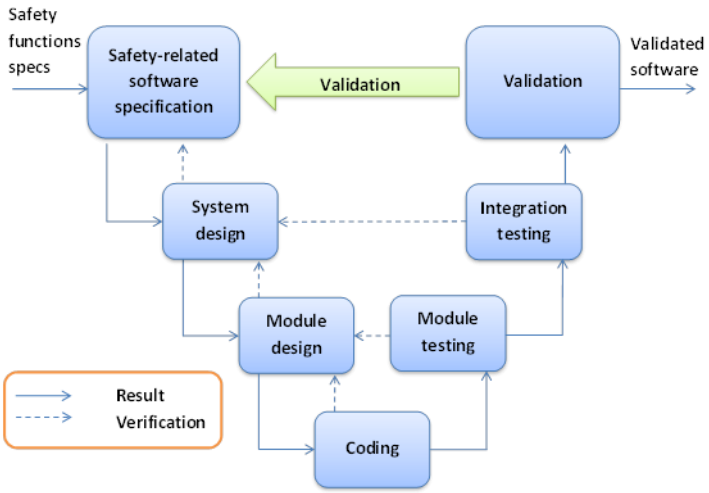
Consequences and severity	S	Class Cl = F + W + P				
		3-4	5-7	8-10	11-13	14-15
Death, losing one eye or arm	4	SIL 2	SIL 2	SIL 2	SIL 3	SIL 3
Permanent, losing fingers	3		(AM)	SIL 1	SIL 2	SIL 3
Reversible, medical treatment	2			(AM)	SIL 1	SIL 2
Reversible, first aid	1				(AM)	SIL 1

So, a safety function designed to protect a user from harm that shows medium probability to happen, with low severity consequences (for example reversible injury with medical treatment required), with frequent (more than one occurance per hour) and long exposure and scarcely or impossible avoidance chances would be categorized as:

- **PLc** according to ISO 13849-1 (S_1 -> F_2 -> P_2 -> PLc)
- **SIL1** according to IEC 61062 (F(5) + W(3) + P(5) = 13 -> S(2) -> SIL 1)

Safety cycle

Considering the 16-phase safety lifecycle defined in IEC 61508 as a reference (figure 2 in IEC 61508-1), in relation to the topic of this paper, Phase 9 "Safety-related systems: E/EE/PES" is the most relevant. Specifically, the requirements for the software safety lifecycle defined as a sub-phase of Phase 9 are described in detail in IEC 61508-3: "Software requirements". Amongst other elements, Part 3 details requirements for software architecture design, support tools (including programming languages), code implementation, integration, testing etc.



Application and embedded software

According to ISO 13849, the software for systems that are meant to implement safety

functions can be:

- SRASW (safety-related application software): software specific to the application, implemented by the machine manufacturer to meet the SRP/CS requirements (ISO 13849 – Ch. 3.1.36);

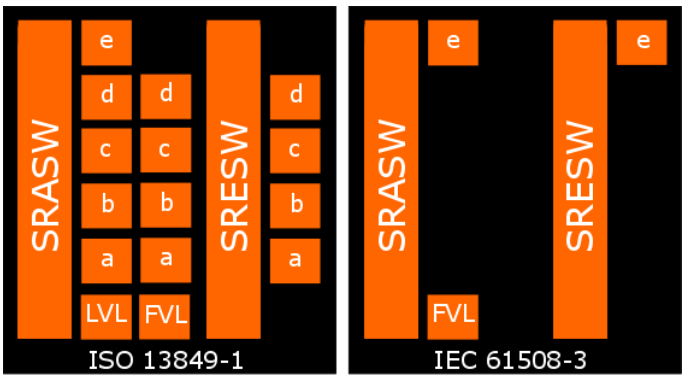
  or

- SRESW (safety-related embedded software): embedded software, firmware or system software that is part of the system supplied by the control manufacturer and which is not accessible for modification by the user of the machinery (ISO 13849 – Ch. 3.1.37).

Software development domain

In accordance to the relevant standards, *programming languages* can be divided into two categories:

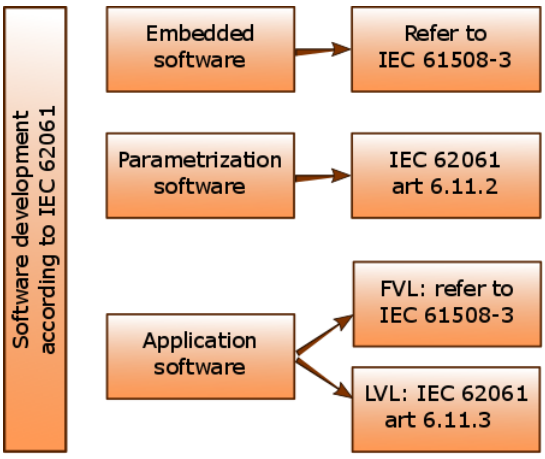
- LVL – limited or low variability language – a language with low complexity and limited performance range, such as function block diagram (FBD) and ladder diagram (LD). (ISO 13849 – Ch. 3.1.34); and
- FVL – full variability language – a language with a complete functional range, such as C, Assembly and so on. (ISO 13849 – Ch. 3.1.34).



ISO 13849 allows for the development of embedded software up to PLd (SIL 2); to achieve PLe (SIL 3) the embedded software development has to be aligned to the requirements of IEC 61508-3.

In terms of application software developed with LVL, such a restriction is not present and the development is allowed for all PL levels under the ISO 13849-1 framework.

The scope of application of IEC 62061 in terms of software development guidelines is rather different. IEC 62061 covers only aspects related with application software implementations (SRASW), in basic or extended levels, using LVL and parametrization software. Whether embedded software (SRESW) or application software is developed with full variability, languages have to be targeted under IEC 62061, the reference is once again IEC 61508-3.



ISO 13849 software requirements

ISO 13849 provides some general specifications for safety related software development; some additional requirements are defined depending on the performance level targeted on the language used and on the type of safety function software developed (SRESW or SRASW).

The safety software lifecycle requirements are defined around a simplified V-model structure. On top of the general requirement to provide complete, available, readable and understandable documentation, modular and structured design and coding, control of systematic failures, functional (black box) testing, LC activities to manage modifications, some additional measures are described:

- SRESW for components with PLr from c to d is required to implement a project management and quality management system comparable with, say, IEC 61508 or ISO 9001. Also requested are configuration management to identify all configuration items and documents related to a SRESW release, structured specification with requirements and design, modular and structured programming (separation in non-safety-related software, limited module sizes with fully defined interfaces, use of design and coding standards), coding verification by walk-through or review with control flow analysis, extended functional testing (grey box testing, performance testing or simulation), impact analysis and appropriate software safety lifecycle activities after modifications.
- for SRASW written in FVL with PLr from a to e, the same requirements as SRESW apply.
- for SRASW written in LVL with PLr from c to e, along with the general requirements, some additional measures are required (with efficiency increasing from PLr of c to PLr of e) with impact in the areas of specifications, tools selection, design, implementation and coding, documentation, verification, configuration management, and modifications.

IEC 61508 software requirements

IEC 61508 is a type A standard; as such, parts 1, 2, 3 and 4 of IEC 61508 are IEC basic safety publications. IEC technical committees should, wherever practicable, make use of these parts in the preparation of their own sector or product standards that have E/E/PE safety-related systems within their scope [3]. This happens in machinery, as already observed, where to achieve the highest level of risk reduction while using full variability languages both ISO 13849 and IEC 62061 require the application to comply with the requirements described in IEC 61508-3. A deeper description of such requirements and how PRQA products can help to satisfy them is the object of part two of this document.

Coding standards

All the standards recognize the importance of *coding standards* and in respect to complying with the most stringent SILs a coding standard is always highly recommended or mandatory. However, it is worth noting that none of IEC 61508 family of standards explicitly states which coding standard to use. ISO 26262 comes closest, highlighting the MISRA C coding standard [4], but it stops short of mandating it. Note that in practice MISRA is standard de facto within the automotive industry as any project opting not to use MISRA would most likely raise eyebrows. MISRA is one of the longest established and most respected of coding standards, with the first revision, MISRA C: 1998 “Guidelines for the Use of the C Language in Vehicle Based Software”, published more than 17 years ago. Additionally it is also important to note that MISRA has been adopted in every market that creates safety critical software. Indeed, the title of the most recent MISRA C: 2012 standard “Guidelines for the use of C language in critical systems” [5] clearly signals the broader scope.

Tools

All modern software development is accompanied by a supporting cast of software tools, from modeling, compiling and debugging to testing and analyzing. With respect to these tools all the standards adopt a very similar approach. They recognize the fact that all tools are not equal, and they define, typically, three classes of tool that are ranked according of the potential impact on the software if the tool malfunctions. All the standards then define sets of criteria that must be met to ensure the tools within each class are fit for purpose.

Summary

The status of machinery safety standard adoption and penetration is bound to change substantially in the near future. Currently there is a noticeable prevalence of ISO 13849 compliant projects; however IEC 62061 is quickly gaining market share. A merging process that will produce the joint standard ISO/IEC 17305 “Design of safety functions realized by control systems” has a final draft (FDIS) expected to be submitted by early 2017. However, software requirements are expected to keep all the relevant aspects that have been characterizing their management in ISO 13849 and IEC 62061.

About PRQA, QA·C / QA·C++ and MISRA

PRQA pioneered coding standard inspection and is recognized worldwide as the coding standards expert due to its industry-leading software inspection and standards enforcement technology. PRQA's QA·C and QA·C++ static analysis tools offer two of the most comprehensive parsers available today, providing detailed information and accurately enforcing coding standards and best practices.

QA·C 8.2.2 with MISRA C (referred to as "QA·C") and QA·C++ 3.2.2 with an extended MISRA C++ (referred to as "QA·C++") have been certified by SGS-TÜV SAAR as fit for purpose to develop safety-related software up to SIL 4 according to IEC 61508 (if used as described in the appropriate Safety Manual). The MISRA C++ extended compliance module adds some additional rules over those in MISRA C++ to meet some of the standard's requirements.

IEC 61508 – Part 3: Software requirements

Part 3 of IEC 61508 addresses the software requirements of a safety-related system, including several tables that define the methods that must be considered to achieve compliance with the standard. As previously mentioned, most of these objectives are also required when the PL/SIL sought for SRESW under development is lower than PLe/SIL3; as a consequence, the same mapping can apply for the relevant activities for IEC 62061 and ISO 13849. The safety manual, QA·C with MISRA-C, also contains all necessary requirements relating to documentation and references to results and validation.

Summary

ISO 13849 and IEC 62061 are the main functional safety standards used in machinery. Both the standards adopt a similar approach to identify risks, harms and to define the level of risk reduction to be achieved by a specific safety function (SIL or PL). Depending on the SIL or PL desired, the requirements of the software process needed to implement the safety functions can be more or less demanding. QA·C and QA·C++ with the MISRA compliance modules have been certified for use with IEC 61508 projects up to SIL4. Thus the time and cost of meeting many of the standard's requirements associated with development at the software level can be reduced by using these tools.

Glossary

E/E/PE	Electrical, Electronic and Programmable Electronic
FBD	Functional Block Diagram
FVL	Full Variability Language
LD	Ladder Diagram
LVL	Low Variability Language
MISRA	Motor Industry Software Reliability Association
PL	Performance Level
PLC	Programmable Logic Controller
SIL	Safety Integrity Level
SRASW	Safety-Related Application Software
SRESW	Safety-Related Embedded Software
SRP/CS	Safety-Related Part of a Control System

References

[1] D. Greenfield, "How Embedded Systems Are Changing Automation," 1 February 2013. [Online]. Available: http://www.automationworld.com/embedded-control/how-embedded-systems-are-changing-automation.

[2] Rockwell Automation, "Introduction to Functional Safety of Control Systems," [Online]. Available: http://www.ab.com/en/epub/catalogs/3377539/5866177/3378076/7555769/tab3.html.

[3] I. E. (. L. a. H. &. S. L. (HSL), "A methodology for the assignment of safety integrity levels (SILs) to safety-related control functions implemented by safety-related electrical, electronic and programmable electronic control systems of machines," 2004. [Online]. Available: http://www.hse.gov.uk/research/rrpdf/rr216.pdf.

[4] IEC, "Guide 104," [Online]. Available: https://webstore.iec.ch/preview/info_iecguide104%7Bed4.0%7Den.pdf.

[5] MISRA - Motor Industry Software Reliability Association, [Online]. Available: http://www.misra.org.uk/Publications/tabid/57/Default.aspx.

[6] PRQA, "MISRA C:2012 PRQA White Paper WP120A/02/13," [Online].

About PRQA

Established in 1985, PRQA is recognized throughout the industry as a pioneer in static analysis, championing automated coding standard inspection and defect detection, delivering its expertise through industry-leading software inspection and standards enforcement technology.

PRQA static analysis tools, QA·C and QA·C++, are at the forefront in delivering MISRA C and MISRA C++ compliance checking as well as a host of other valuable analysis capabilities. All contain powerful, proprietary parsing engines combined with deep accurate dataflow that deliver high fidelity language analysis and comprehension. They identify problems caused by language usage that is dangerous, overly complex, non-portable or difficult to maintain. Additionally, they provide a mechanism for coding standard enforcement.

Contact Us

PRQA has offices globally and offers worldwide customer support. Visit our website to find details of your local representative.

Email: info@programmingresearch.com
Web: www.programmingresearch.com