

# Problems in the IEC 61511 Standard

Arian Slagt  
Yokogawa System Center Europe

## Abstract

This paper will have a close look at the IEC 61511 safety standard from the practical SIS supplier's point of view. The paper may look a bit negative with respect to the standard. That is not my intention. My intention is to contribute to the improvement of the standard, since it is in nature a very good guideline for the process industry.

Specifically the clause on Safety Requirement Specifications (SRS) and the clause on SIS Application Software will be examined. Regarding the SRS it appears that the standard is not clear, and is also mixing the requirements with the design. The conclusion on the SIS application software part must be that it is rather overdone with many impossible requirements.

At the end of this paper recommendations to improve the standard are given.

Although the standard does not mention the different parties involved in the SIS lifecycle and their responsibilities, these parties will have impact on the usability of the standard. This paper assumes the case that the end-user or contractor has written a specification for a SIS, to be delivered by the SIS supplier. Obviously this should be the normal way of doing business in the process industry.

## 1. Safety Requirement Specification

Clause 10 of the standard deals with the so-called Safety Requirement Specification (SRS). In here all items that must be covered by the SRS are mentioned. In general the requirements for the safety instrumented functions must be specified, and this information shall be sufficient to build the SIS. The SIS is defined in the standard to include the sensors, final elements and the logic solver including hardware, software and the application program. For the specification of the application program the additional requirements as per clause 12.2.2 must be obliged.

So far the standard is clear, although it would have been more logical to include all requirements on the SRS together in one chapter.

But there is one more requirement regarding the SRS: clause 10.3.2 says: "the software safety requirement specification shall be derived from the SRS and the chosen architecture of the SIS" This is a very unclear requirement.

First a new notion is introduced: "software SRS". Is this the same as the "application software SRS" from clause 12.2.2? Then why the new definition? Is some other software meant? That is not possible according to clause 12 where the scope of the standard is limited to application software. Let us assume that application software is meant.

Further the "chosen architecture of the SIS" imposes another problem. When the architecture is chosen already by the end-user or contractor it should be part of the SRS itself. But then it is useless to mention it separately in clause 10.3.2. When the chosen architecture must be defined by the SIS supplier, it is impossible for the end-user to prepare the SRS as it is partly based on a future document. In that case is not a requirement, but it should be part of the SIS design.

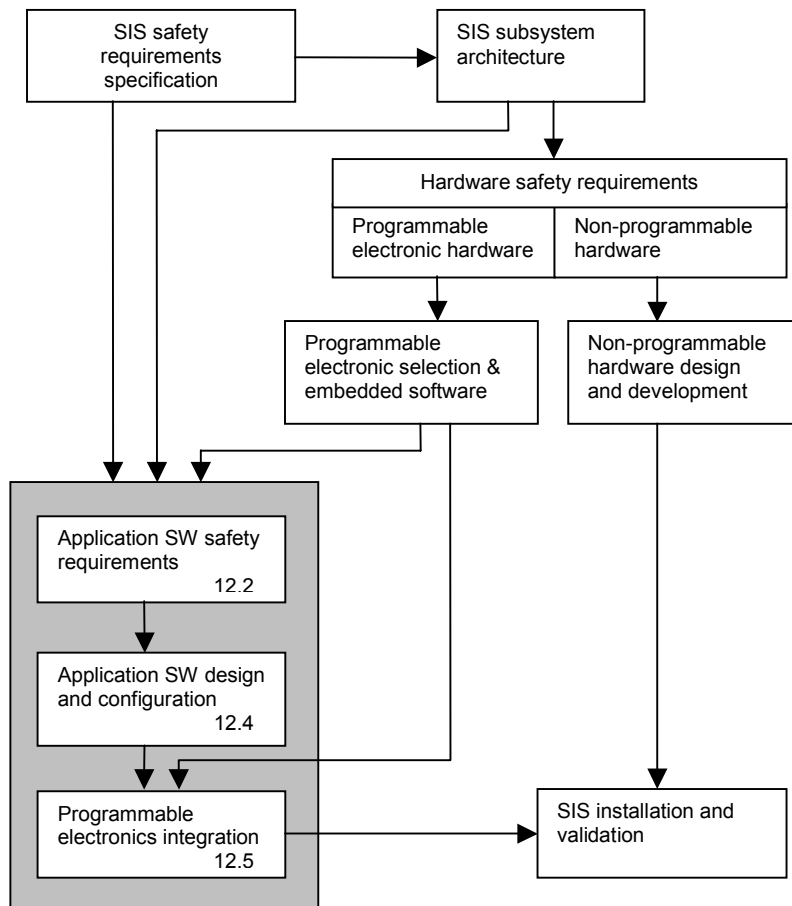
It must be concluded that clause 10.3.2 is impossible to use and therefore it is advised to remove it from the standard.

## **SIS design**

When reading further in the standard, it is described how to design and engineer the SIS. In clause 11.2 it is stated that the design of the SIS should be based on the SRS, which is not really a surprise. But again what is a SIS? The standard shows that the SIS includes both hardware (HW) and software (SW). That means that this design as indicated here must include the software design. This is in contradiction with clause 12, which describes the application software requirements and design separately.

Note also that in this clause no architecture of the SIS hardware is mentioned. Also no "hardware safety requirements" are mentioned. Either it is defined somewhere later on (which is not logical), or it is obviously assumed to be part in the design process (which sounds logical).

Reading the next chapter of the standard: "requirements for application software", it starts with a picture (figure 10) that tries to explain the relation between hardware / application software / requirements / design. This figure has been copied below.

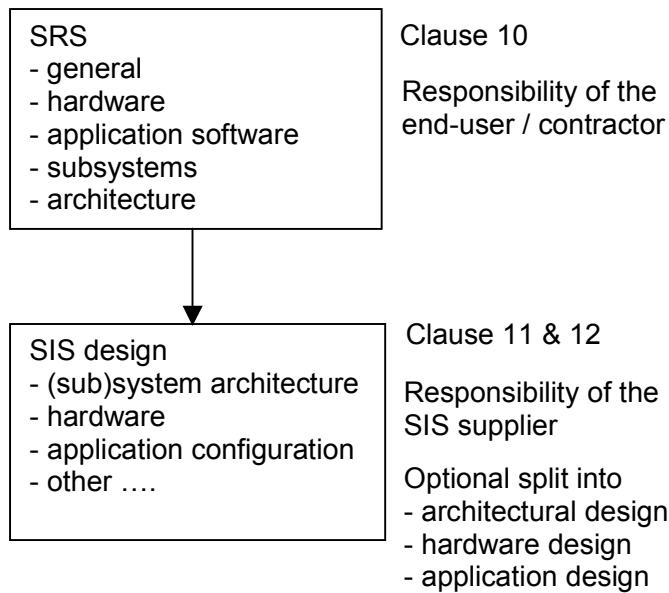


(Copy of figure 10 of IEC 61511 standard)

When looking at the figure it shows that so-called application software safety requirements are derived from the SRS, the SIS subsystem architecture and the programmable electronics selection. Although not indicated it seems that the latter is covered by clause 11.

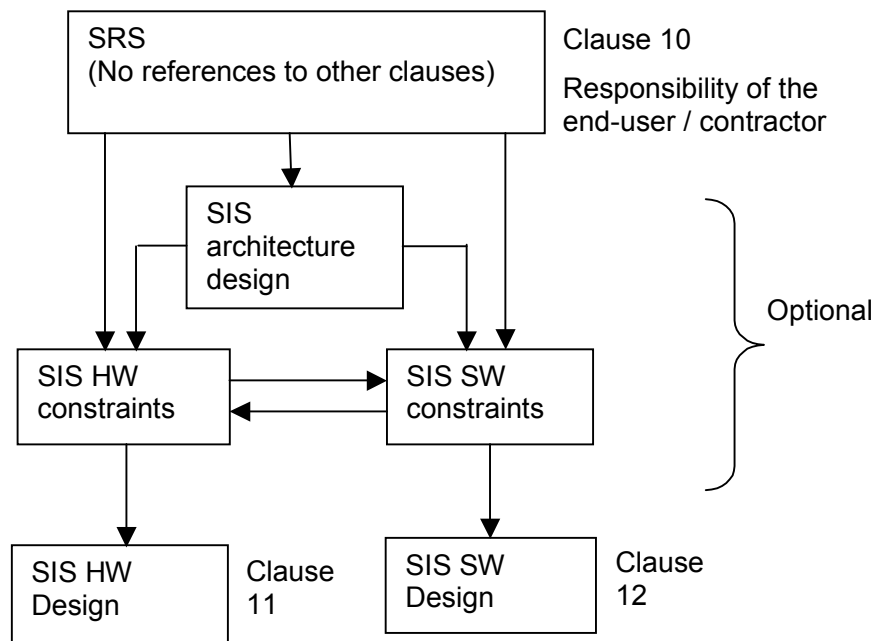
The "SIS subsystem" is not defined in the standard. There is only a kind of explanation in the note (\*) of figure 10. Figure 10 also shows the "hardware safety requirements". This looks like a new and unknown document, which is not mentioned anywhere else in the standard. Or is it a part of the SRS? In that case also the SIS subsystem architecture must be a part of the SRS. Or are all these items part of the SIS design as prepared by the SIS supplier?

Assuming that the SIS architecture is part of the SRS as per clause 10, the figure below gives a much simpler and unambiguous relation between SIS requirements, architectures and design documents:



Clause 12.2.2.4 says that the deficiencies in the application software SRS must be identified to the "SIS subsystem developer". This suggests that this guy has written the application software SRS. That means that it is not part of the SRS from the end-user, but is prepared later on. But then the standard should mention the design of the hardware- and software-architecture as an activity somewhere.

That shows again an unbalance in the standard: why is it required to prepare only an application SW safety requirement and not also a HW safety requirement?  
It would be more consistent to modify figure 10 of the standard as follows:



When the SRS is complete, the optional steps to specify an architecture and to develop additional constraints can be omitted. Both the HW and the SW design can be based directly on the SRS. This is the preferred way.

When the SRS is not complete or not detailed enough the additional steps have to be executed. Care has to be taken that the additional requirements are not in contradiction with the original end-user requirements as specified in the SRS.

This figure should be placed in the beginning of the chapter on SIS design and engineering of course.

## 2. Application Software

The IEC 61511 standard limits its scope to application software developed using Fixed Program Languages (FPL) and Limited Variability languages (LVL). Both will be discussed in detail in the next paragraphs.

### Fixed Program Language (FPL)

FPL will mostly be used for the programming of field devices. For example a transmitter in which the ranges can be set manually or a valve positioner in which the stroke test interval can be set via the HART protocol.

What can be "programmed" in such a device?

- the ranges are specified in the Safety Requirement Specification
- the way to set a specific value is defined by the manufacturer and described in the safety manual of the device
- during the final validation the actual instrument settings are validated against the specification

This has nothing to do with "programming" in the usual sense of the word; it only has to do with limited configuration. It is a selection of the limited possibilities of the device.

Is there really a need for an application software lifecycle to do this configuration work? Including a V-model? Most of the requirements of clause 12 are not feasible in case of FPL devices.

Example: for most devices on the market today there is no choice between FPL and LVL: transmitters and valves are FPL. This means that clause 12.4.2.3 has no practical value, it is impossible to make the required selection between these two languages. The selection is already dictated by the manufacturer of the device.

Another example: 12.4.4.7 "the safety manual (of the device) shall address the following items..." When the transmitter is an IEC 61508 certified one, it must have a safety manual that complies with that standard. And not to this clause.

Therefore it is recommended to use only clause 11 (SIS design and engineering) for all devices with FPL.

Note that "smart sensors" are already mentioned in this chapter.

### **Limited Variability Languages (LVL)**

In the process industry most SIS's are based on an IEC 61508 certified safety PLC. Included in this PLC is a certified set of engineering and supporting tools. Most of these engineering tools are based on IEC 61131-3 LVL language: application configuration is done in Ladder Diagrams, Function Blocks or Sequential Function Charts.

One property (drawback?) here of is the fixed relation between the PLC and its tools. When a contractor buys a PLC of brand Y, he has to use the related tools of brand Y. The only choice left to him is between Ladder Diagrams, Function Blocks or Sequential Functions Charts. I do not expect that this will change much in the coming decade.

So when the objective of the standard is that one has to select the best set of tools (12.4.1.3) it is an impossible mission. Also requirements on the selected application language (12.4.2.3) are defined by the manufacturer of the engineering tools and not really selectable by the user. There are more sub-clauses that are more or less superfluous e.g. clause 12.4.4.4 and 12.4.4.7 as the requirements in there are covered by the IEC 61508 certification of the logic solver already.

When taking these facts into account, and also the discussion on the SRS as above, the main part of clause 12 is not really useful when using certified PLC's with their programming tools.

Looking from some distance to the IEC 61511 standard one gets the feeling that the application software part still looks very similar to the same part of the IEC 61508. Most figures and tables seem to be copied into the IEC 61511. The IEC 61508 is specifically intended for developments using Full Variability Languages like C++ and Java, while the IEC 61511 very clear excludes these languages. But it seems that a lot of details that are needed for FVL only still remain in the IEC 61511. Maybe this is one of the reasons that the application software part of the IEC 61511 is overdone, compared to the other parts of the standard.

Moreover parts of the chapter on application software already point to the overall lifecycle:

clause		Point to :
12.3	SW safety validation planning	Refers completely to clause 15
12.4.6	Module testing	Refers to 12.7 + some additional items
12.4.7	Module integration testing	Refers to 12.7
12.6	Modification procedures	Refers to 5 and 17 + some additional items
12.7	SW verification	Refers to 7 + some additional items

For the execution of safety projects it would have been much easier if hardware engineering and application engineering would have been combined into one set of rules.

Is there really a need for an overall lifecycle and a separate additional application software lifecycle? Is the hardware so simple that it not deserves its own lifecycle? It is strongly advised to integrate the additional requirements for the application design and engineering phase into the overall lifecycle of clause 6.

### 3. Conclusion and recommendations

The conclusion of this paper is that the current version of the IEC 61511 standard is not unambiguous and not clear on the subjects of the SRS and the application software lifecycle. Therefore it is advised to the users of the standard:

1. Include in your own Functional Safety Management system the definitions as you use them in your organization.
2. Define in your FSM that you will follow the lifecycle as given in chapter 6 of the standard. Integrate the requirements for application software into your FSM, so that one integrated lifecycle is left in your project execution.

To the coming IEC 61511 maintenance committee it is proposed to change the following items, and to revise the structure of the standard accordingly in the coming maintenance cycle. This will make the standard more consistent, unambiguous and easier to use.

1. Give a clear definition of the used terms e.g. SRS, software SRS, application software SRS, hardware safety requirements, etc. Explain if it is part of the specification or part of the design.
2. Give one clear figure in the beginning of the clause on SIS design to explain the relation between hardware and application software, and between requirements and design.
3. Remove the split between the hardware and the application software lifecycle: one integrated lifecycle will do.
4. Use clause 11 for FPL devices.
5. Limit the use of IEC 61511 to certified safety PLC's.
6. Modify the structure of clause 12 (and some other clauses) as follows:

Clause	Name	Proposed Revision
7	Verification	Add specific SW requirements from 12.7
10	SRS	Include requirements from 12.2.2 Delete 10.3.2 Add a note that the architecture should be included as needed
11		Add a figure. Add a note that the application configuration is covered separately.
12	Requirement for application software	FPL : to apply clause 11 LVL : limit scope to certified PLC's FVL, SIL4 : to adhere to IEC 61508
12.1	SW safety lifecycle requirements	Delete, integrate into clause 6
12.2	SW safety requirement specification	Delete, is included in clause 10
12.3	SW safety validation planning	Delete, is referring to clause 15 already
12.4	SW design and development	Ok
12.4.1	Objective	
12.4.2	General	Describes the architecture and design To be extended
12.4.3	Architecture	Delete, covered by clause 10 and/or 12.4.2
12.4.4	Tools, user manual, languages	Ok
12.4.5	Application software development	Ok
12.4.6	Module testing	Revise : refer to 7 + some additional items
12.4.7	Module integration testing	Revise : refer to 7 + some additional items
12.5	SIS Integration testing	Ok
12.6	Modification procedures	Delete : covered by 17
12.7	SW verification	Delete : covered by 7
17	SIS modification	Add specific requirements for application software (if any)