# Position Paper on IEC 61508 2010 Definitions Regarding Minimum Hardware Fault Tolerance / Architectural Constraints

The release of IEC 61508 2010 has led to several discussions on how certain new, updated, and unmodified definitions need to be interpreted. The controversy relates to the determination of the required minimum hardware fault tolerance / architectural constraints interpretation.

This position paper explains the position that exida has taken with regard to this issue. The position paper is structured in two parts; the position and the Rationale for the position including counter arguments received over the last couple of months. The exida position is also implemented in the exida exSILentia safety lifecycle tool.

## 1 Positions

### 1.1 Use of standard

End-users / owners / operators in the Process Industries should seek compliance with IEC 61511 (2003) or ANSI/ISA 84.00.01-2004 (IEC 61511 Mod). The minimum hardware fault tolerance / architectural constraints are addressed in section 11.4. exida suggest users to follow clause 11.4.5 that allows users to deviate from the concepts defined in 61511 as long as they are compliant with the concepts as defined in IEC 61508-2, Tables 2 and 3 [Rationale 1]. The reference is towards the 2000 edition of 61508 and should not be interpreted as a future reference to IEC 61508 2010.

### 1.2 Element

IEC 61508 2010 introduces the element concept in part 4 Clause 3.4.5. The minimum hardware fault tolerance requirements (SFF and Type definition) are applied on a per element basis. An element is defined as:

> ***element***
> *part of a subsystem comprising a single component or any group of components that performs one or more element safety functions.*
> *NOTE 1 An element may comprise hardware and/or software.*
> *NOTE 2 A typical element is a sensor, programmable controller or final element.*

It is exida's position that an element should contain all equipment/devices that are needed to perform a safety function [Rationale 2].

## 1.3 Failure Mode Definitions

Though IEC 61508 2010 acknowledges that there are more failure modes than simply dangerous and safe the definitions lack clarity. It is exida's position that the following base definitions should be used [Rationale 3].

### 1.3.1 Fail Dangerous

A dangerous failure is defined as a failure that does not respond to a demand from the process (i.e. being unable to go to the defined fail-safe state).

### 1.3.2 Fail Safe

A safe failure is defined as a failure that results in the presentation of the selected fail-safe input or output condition without a demand from the process.

### 1.3.3 Fail Annunciation

An annunciation failure is defined as a failure that has no effect on the safety function but does affect the ability to detect future faults, for example a failure of an internal diagnostic circuit of an equipment item.

### 1.3.4 Fail No Effect / Residual

The No Effect / Residual failure category represents failures of components that are part of the safety function but that have no effect on the correct functioning of the safety function. Note that equipment item components that are not part of the safety function (but may be part of the product design) and that cannot affect the safety function are not included in this category.

### 1.3.5 Detected

In relation to hardware, detected by automatic diagnostic tests, internal or by a connected safety logic solver

### 1.3.6 Undetected

In relation to hardware, undetected (not diagnosed) by automatic diagnostic tests, internal or by a connected safety logic solver

### 1.3.7 Revealed

In relation to hardware, detected by proof tests, operator intervention (for example physical inspection and manual tests), or through normal operation

### 1.3.8 Unrevealed

In relation to hardware, undetected (not diagnosed) by proof tests, operator intervention (for example physical inspection and manual tests), or through normal operation

## 1.4 Safe Failure Fraction (SFF)

It is exida's position that the SFF should be calculated as follows [Rationale 4].

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

# 2 Rationales

## 2.1 Rationale 1: Use of standard

The simplified minimum hardware fault tolerance concept in IEC 61511 is in general more stringent than the concepts as defined in the 2000 edition of IEC 61508-2 table 2 and 3. For example IEC 61511 will require that a final element without proven in use justification has a hardware fault tolerance of 1 (given the fact that the industry average final element does NOT have a dominant failure mode that is to the safe state nor are dangerous failures detected 11.4.3). If proven in use can be claimed (note that control operation does not count towards safety proven in use) the minimum hardware fault tolerance is 0. There is no credit for good engineering practices in this approach by using for example partial stroke testing (which would help detect stuck stem type of failures).

## 2.2 Rationale 2: Element

The following are examples of what an element would be per exida's position

1. If a "transmitter" is used to perform a process measurement and communicate the measurement value to a logic solver, than the element is defined as the combination of any impulse lines or other process connections, the sensing element(s), and the transmitter.
2. If a "transmitter" and "isolation barrier" are used to perform a process measurement and communicate the measurement value to a logic solver in an isolated/intrinsically safe manner, than the element is defined to include in addition to the devices listed for the previous example also of the isolation barrier.
3. If a solenoid is the final element, i.e. it performs the final safety action, the (final) element only consists of the solenoid.
4. If a solenoid operates an actuator which operates a valve, than the safety function is only accomplished by the combination of these three devices and the element is defined as the combination of the solenoid, actuator, and valve.

The following statement was made by a member of the IEC 61508 committee who was involved in both the 2000 and 2010 release, the statement relates to a discussion if the solenoid device type should dictate the entire final element device type.

> *"An element or group of elements can perform an 'element safety function' (Pt 4 Cl 3.3.6), and the definition of type A or type B is done at the 'element' level, not at the 'element safety function' level. Consequently it would be consistent with the requirements of the standard if the SOV was considered as a separate element, forming a "channel" with the Actuator, which together perform the 'element safety function' of "responding to closure command and implementing executive action."*
>
> *I can see no justification from the text of the standard (IEC61508 Ed 2) for insisting that the Type classification of the SOV be used to determine the Type classification of the final actuator, and it is illogical to do so.*

The counter argument for this statement is the following:

*The justification is in the basis of the standard, the definitions. It is not an interpretation of the 2nd edition of the standard; it is based on explicit statement in the standard as to what an element is. Consider the actuator instead of the solenoid. There is no case to be made to define the actuator as an element in the above example. The actuator does not perform a complete function, it needs the solenoid, and it needs the valve. Extending this reasoning results in the conclusion that the solenoid cannot be defined as a single element.*

*If one wants to argue that the actuator is an element, than one would have to take that approach all the way down to the individual components as these cannot fulfill complete functions either. In that scenario the spring would be an element as well. This is illogical.*

Based on this an element must contain all equipment/devices that are needed to perform a safety function.

## 2.3 Rationale 3: Failure Mode Definitions

The main ambiguity in IEC 61508 2010 with relation to the definition of failure modes relates to the definition of detected and undetected failures. The standard currently states in part 4:

***3.8.8***
***detected***
***revealed***
***overt***
*in relation to hardware, detected by the diagnostic tests, proof tests, operator intervention (for example physical inspection and manual tests), or through normal operation*
*EXAMPLE These adjectives are used in detected fault and detected failure.*
*NOTE A dangerous failure detected by diagnostic test is a revealed failure and can be considered a safe failure only if effective measures, automatic of manual, are taken.*

***3.8.9***
***undetected***
***unrevealed***
***covert***
*in relation to hardware, undetected by the diagnostic tests, proof tests, operator intervention (for example physical inspection and manual tests), or through normal operation*
*EXAMPLE These adjectives are used in detected fault and detected failure.*

The problem with these definitions is that faults detected by proof tests, operator intervention, or normal operation are defined as detected failures. If this definition would be used to determine the "dangerous undetected" failure rate then the formula to calculate this would be:

$$\lambda_{DU} = \lambda \cdot (1 - \%Safe) \cdot (1 - DC) \cdot (1 - PTC) \cdot (1 - OIC) \cdot (1 - NOC)$$

Where:

- %Safe = Percentage of safe failures
- DC = Diagnostic Coverage

- PTC = Proof Test Coverage
- OIC = Operator Intervention Coverage
- NOC = Normal Operation Coverage

Even if the operator intervention and normal operation detection would be ignored, these definitions suggest that the proof test coverage has an impact on the dangerous undetected failure of an element. This is an incorrect approach;

1. Proof test effectiveness, completeness and correctness, should be taken into consideration at a system modeling level.
2. SFF is an element property and should not be dependent on implementation.

## 2.4 Rationale 4: Safe Failure Fraction (SFF)

The Safe Failure Fraction is a simple ratio of safe and dangerous detected failure to the total failure rate. Annunciation and No Effect / Residual failures are not included in the SFF calculation.

*This was the original exida approach in 2000. The only reason Annunciation and No Effect / Residual failures were included in the SFF calculation was the poor definition of safe failures in IEC 61508 2000 edition which declared all non dangerous failures safe.*

Proof tests, operator intervention, normal operation do not impact the calculation of the SFF.

*This is a logical result from the definitions as maintained by exida*

Second order failures (a detectable dangerous failure occurring after diagnostic circuitry failure) are excluded from the SFF calculation.

*Second order failures should be accounted for in system modeling not in element properties.*

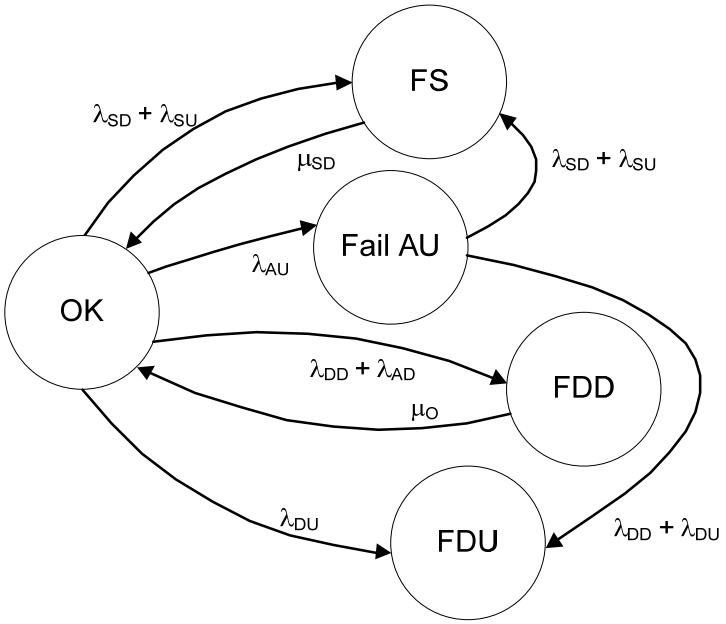
$$SFF = \frac{\lambda_{SD} + \lambda_{SU} + \lambda_{DD}}{\lambda_{SD} + \lambda_{SU} + \lambda_{DD} + \lambda_{DU}}$$

Alternate notations:

$$SFF = 1 - \frac{\lambda_{DU}}{\lambda_{SD} + \lambda_{SU} + \lambda_{DD} + \lambda_{DU}}$$

$$SFF = \%Safe + (1 - \%Safe) \cdot C_D$$

*An example Markov Model on how loss of automatic diagnostics (annunciation failures) can be taken into consideration is show below.*


FS
Fail AU
OK
FDD
FDU
λSD + λSU
μSD
λSD + λSU
λAU
λDD + λAD
μO
λDU
λDD + λDU