Safety Primer

Safety basics for complex systems - Terms and Standards

Nicholas Mc Guire <safety@osadl.org>
<mcguire@lzu.edu.cn>

July 5, 2019

Safety Critical Linux Working Group

Outline

Functional safety - like any sound engineering discipline - can only work if we have a consistent set of terms and processes. This first session will focus on that context:

- A few basic terms to get started
- A high-level view on safety standards

Nicholas Mc Guire
<safety@osadl.
<mcguire@lzu.

Basic Definitions

- **hazard**:
  potential source of harm
  [IEC 61508-4 Ed 2 3.1.2]
- **risk**:
  combination of the probability of occurrence of harm and the severity of that harm
  [IEC 61508-4 Ed 2 3.1.6]
- **tolerable risk**:
  risk which is accepted in a given context based on the current values of society
  [IEC 61508-4 Ed 2 3.1.7]
- **safety**:
  freedom from unacceptable risk
  [IEC 61508-4 Ed 2 3.1.11]

Nicholas Mc Guire
<safety@osadl.
<mcguire@lzu.

Context of safety

- **functional safety**:
  part of the overall safety relating to the EUC and the EUC control system that depends on the correct functioning of the E/E/PE safety-related systems and other risk reduction measures

  [IEC 61508-4 Ed 2 3.1.12]
- **environment**:
  all relevant parameters that can affect the achievement of functional safety in the specific application under consideration and in any safety lifecycle phase

  [IEC 61508-4 Ed 2 3.2.2]
- **reasonably foreseeable misuse**:
  use of a product, process or service in a way not intended by the supplier, but which may result from readily predictable human behavior

  [IEC 61508-4 Ed 2 3.1.14]

Nicholas Mc Guire
<safety@osadl.
<mcguire@lzu.

Safety
a system property

- Is a silent malloc() failure of libc critical ?

Nicholas Mc Guire
<safety@osadl.
<mcguire@lzu.

Safety
a system property

- Is a silent malloc() failure of libc critical ?
- You can't say unless you know:
  - where and when it is being used, and
  - what consequence must be expected if it fails, and
  - what protection (implicit or by design) are in place

Nicholas Mc Guire
<safety@osadl.
<mcguire@lzu.

Safety
a system property

- Is a silent malloc() failure of libc critical ?
- You can't say unless you know:
  - where and when it is being used, and
  - what consequence must be expected if it fails, and
  - what protection (implicit or by design) are in place

For very simple systems we had a trick - we simply assumed everything that can go wrong **is** catastrophic **and** we actually know everything that can go wrong. With that in place we can design an **exhaustive** mitigation..

Nicholas Mc Guire
<safety@osadl.
<mcguire@lzu.

Safety
a system property

- Is a silent malloc() failure of libc critical ?
- You can't say unless you know:
  - where and when it is being used, and
  - what consequence must be expected if it fails, and
  - what protection (implicit or by design) are in place

For very simple systems we had a trick - we simply assumed everything that can go wrong **is** catastrophic **and** we actually know everything that can go wrong. With that in place we can design an **exhaustive** mitigation..
But: This solution does not scale.

Nicholas Mc Guire
<safety@osadl.
<mcguire@lzu.

Type-A/Type-B
The simple and the ugly

- Type-A systems [IEC 61508-2 Ed 2 7.4.1.2]
  - All failure modes known, and
  - Behavior of system under failure condition understood, and
  - Adequate data available to support failure rates of elements.
- Type-B [IEC 61508-2 Ed 2 7.4.1.3]
  - If at least one of the Type-A requirements are not met.

Nicholas Mc Guire
<safety@osadl.
<mcguire@lzu.

Type-A/Type-B
The simple and the ugly

- Type-A systems [IEC 61508-2 Ed 2 7.4.1.2]
  - All failure modes known, and
  - Behavior of system under failure condition understood, and
  - Adequate data available to support failure rates of elements.
- Type-B [IEC 61508-2 Ed 2 7.4.1.3]
  - If at least one of the Type-A requirements are not met.

- Type-A system **can** be verified by testing
- Type-B system **can only** be verified by analysis, testing and monitoring
- Safety process goal: translate Type-B to Type-A systems

Nicholas Mc Guire
<safety@osadl.
<mcguire@lzu.

What standard ?

- When can domain standards be used
  - Unchanged technology compared to SotA of Standard
  - Evolutionary deviation from assumed environment
- What to do if a domain standard is not available ?
  - Use a generic or basic safety standard (61508)
- What to do when the domain-standard is not applicable ?
  - Go up the chain -> basic safety standard (61508)

Nicholas Mc Guire
<safety@osadl.
<mcguire@lzu.

What standard ?

- When can domain standards be used
  - Unchanged technology compared to SotA of Standard
  - Evolutionary deviation from assumed environment
- What to do if a domain standard is not available ?
  - Use a generic or basic safety standard (61508)
- What to do when the domain-standard is not applicable ?
  - Go up the chain -> basic safety standard (61508)

So can we use and existing domain standard like ISO 26262 for autonomous vehicles ?

Nicholas Mc Guire
<safety@osadl.
<mcguire@lzu.

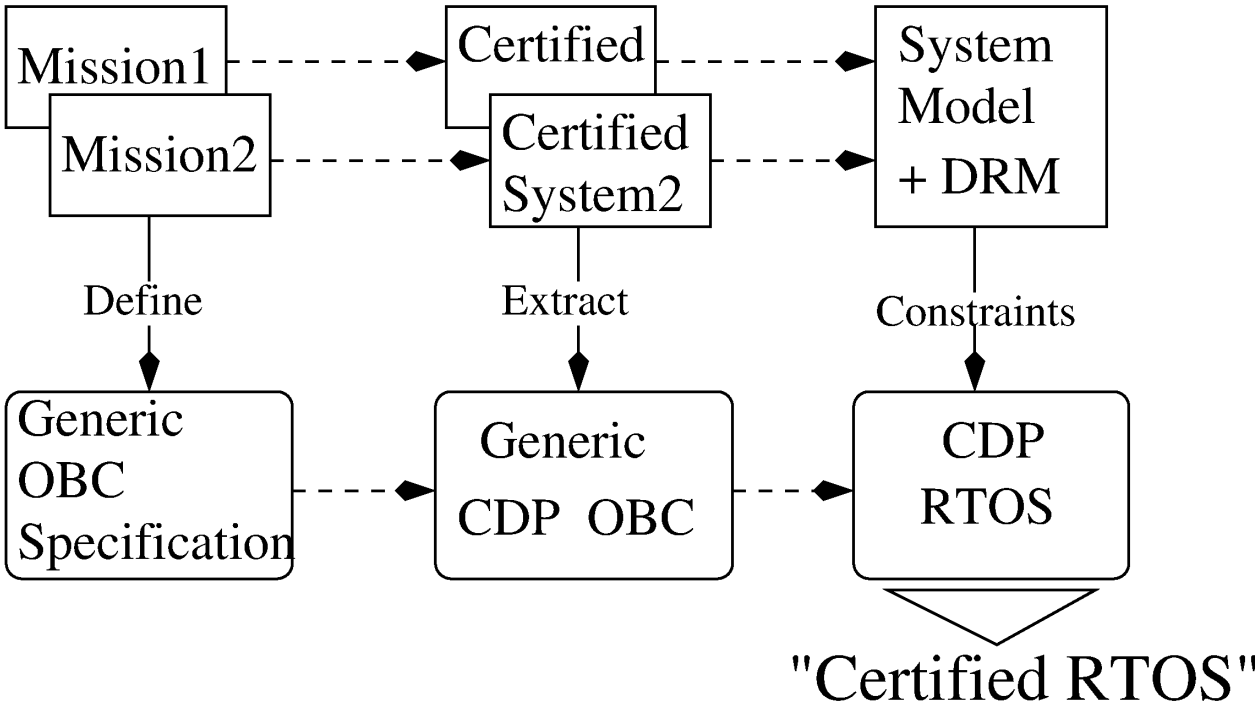
Domain standards

- Encode Context -> reduced analytical scope
- Constrain Complexity -> defined technological boundaries/state-of-the-art
- Common body-of-knowledge as implied basis -> pre-selected target measures and techniques

Domain standards are based on consolidation of a domain with respect to the safety related functionality/products.
They form a regulatory "baseline" that ideally maps to a domain "baseline" with respect to requirements and sometimes design "patterns".

Nicholas Mc Guire
<safety@osadl.
<mcguire@lzu.

"Pre-certified element"



- OBC: On-Board Computer
- DRM: Design Reference Mission
- CDP: Certification Data Package

Nicholas Mc Guire
<safety@osadl.
<mcguire@lzu.

ISO 26262
...short rant only

- Not a very well written standard
- Give little (too little) guidance on the objectives
- Assumes human operated vehicle
- Designed for low-complexity MCUs and
- Low-complexity OSEK class OS

ISO 26262 (as all other 61508 derivatives) are applicable for low-complexity or Type-A systems - but not for highly complex novel-technology based large systems - with other words don't even try to apply it to autonomous vehicles. Why ?

Nicholas Mc Guire
<safety@osadl.
<mcguire@lzu.

ISO 26262
...short rant only

OSADL

- Not a very well written standard
- Give little (too little) guidance on the objectives
- Assumes human operated vehicle
- Designed for low-complexity MCUs and
- Low-complexity OSEK class OS

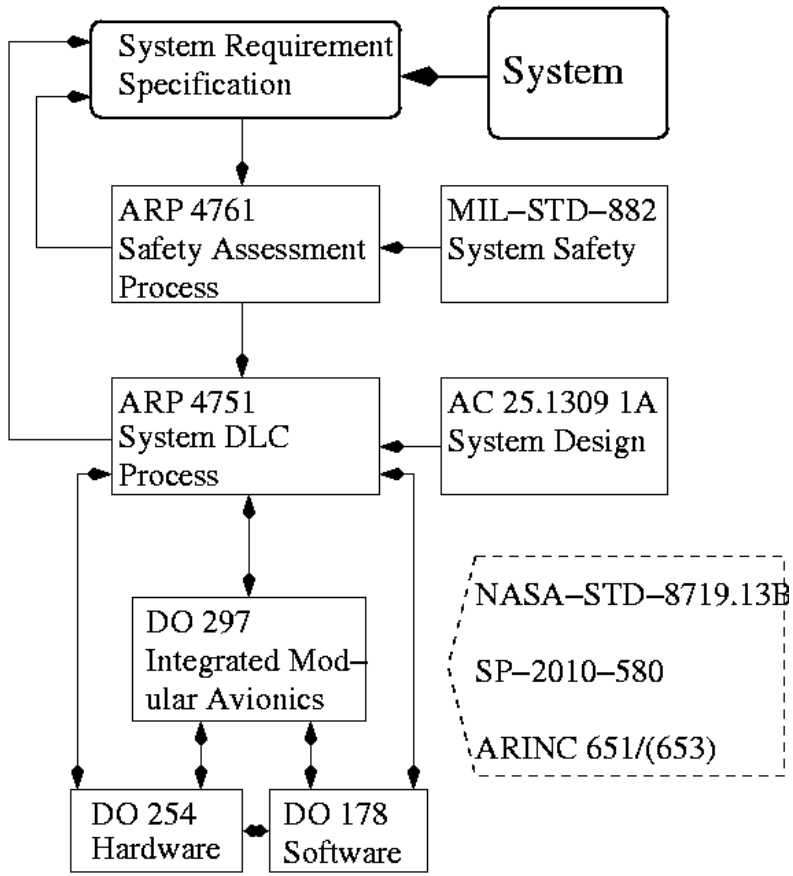
ISO 26262 (as all other 61508 derivatives) are applicable for low-complexity or Type-A systems - but not for highly complex novel-technology based large systems - with other words don't even try to apply it to autonomous vehicles. Why ? Because you **can't** achieve compliance if the standard does not fit your system.

Nicholas Mc Guire
<safety@osadl.
<mcguire@lzu.

Standard Context

...standards are not out-of-context

- Domain standards are embedded in a set of standards.
- The validity of domain standards depends on managing this relation.
- Using a domain standard out-of-context may invalidate it and result in systematically unsafe systems.



Nicholas Mc Guire
<safety@osadl.
<mcguire@lzu.

Compliance
...not certification...

Compliance has two basic parts to it. Satisfaction of:

- Objectives: Objectives must be satisfied
- Requirements: if applicable (and non-applicability shall be justified)

To achieve this - Objectives need to be interpreted **in context**. This happens by **tailoring of the DLC/SLC** for the given system and interpretation of each clause/subclause in context of the specifics of the system (requirements/technologies/competency)

Nicholas Mc Guire
<safety@osadl.
<mcguire@lzu.

Compliance Routes (IEC 61508 Ed2)

- Hardware / random faults:
  - 1_H: Fault-tolerance + SFF
  - 2_H: component reliability data + field data (specified SIL)
- Software / systematic faults:
  - 1_S: Compliant development
  - 2_S: Proven-in-Use
  - 3_S: Assessment of non-compliant development

No 3_H currently defined (one of the SIL2LinuxMP project proposal)
Route 3_S somewhat inconsistent and partially buggy - needs extensive interpretation
Note that basically all domain standards reference you back to IEC 61508 for complex elements.

Nicholas Mc Guire
<safety@osadl.
<mcguire@lzu.

Why 61508 ?

- Targeting novel technologies, large systems, complex systems
- Designed to allow accommodation of innovative changes - with other words 61508 is relatively flexible
- Compliance routes based on objectives
- Clear guidance based on properties to be achieved
- Intended to serve as a basis for domain standards - e.g. 61508 might be a suitable basis for a FLOSS FuSa standard.
- Context-agnostic baseline standard.

Autonomous systems -> IEC 61508 Ed2 (taking the upcoming Ed 3 into consideration)

Nicholas Mc Guire
<safety@osadl.
<mcguire@lzu.

61508 Part 1
..not even a nutshell

OSADL

- It's about managing complexity by designing adequate processes
- Guidance on developing adequate context - its all about context
- Place a focus on eliminating hazards rather than on mitigation.
- Safety **is** a system property - there is no such thing as SEooC and
  - For Type-B elements there never will be.

Nicholas Mc Guire
<safety@osadl.
<mcguire@lzu.

61508 Part 1
..not even a nutshell

OSADL

- It's about managing complexity by designing adequate processes
- Guidance on developing adequate context - its all about context
- Place a focus on eliminating hazards rather than on mitigation.
- Safety **is** a system property - there is no such thing as SEooC and
  - For Type-B elements there never will be.
- The key problems - System evolution analysis
  - "Type-C" - Complex/Autonomous systems undergoing dynamic change ?
  - Knowing dependencies - making change manageable
  - Deriving severity - making impact judgement feasible

Nicholas Mc Guire
<safety@osadl.
<mcguire@lzu.

Summary Notes

- There is no really suitable standard for complex safety related systems at present - the best is still 61508 Ed 2 (with considerations for Ed 3)
- Safety for complex elements needs a lot of interpretation of the standards - you can not follow the standard - you can at best take it as guidance.
- Safety is a system property and for novel technologies one can not work "out-of-context"
- If GNU/Linux, or a subset of it, can be certified is an open question and most likely there is no generic answer but only a context specific answer.

If you want to do safety - start with reading IEC 61508 Ed 2 part 0-3,5,6 with part 4 and 7 on your desk.

Nicholas Mc Guire
<safety@osadl.
<mcguire@lzu.

Accronyms

- AC - Advisory Circular (FAA)
- ARINC - Aeronautical Radio Incorporated
- ARP - Aerospace Recommended Practice
- GNU - GNU Not Unix
- IEC - International Electrotechnical Commission
- ISO - International Organization for Standardization
- DLC - Development Life Cycle
- FLOSS - Free/Liebre Open Source SW
- MCU - Mikrocontroller
- OSEK - **O**ffene Systeme und deren **S**chnittstellen fr die
- SEooC - Safety Element out of Context
- SFF - Safe Failure Fraction
- SIL - Safety Integrity Level

Nicholas Mc Guire
<safety@osadl.
<mcguire@lzu.