

IEC 61508 Viewpoint on System Safety in the Transport Sector

Part 4 — A Safety Assurance Perspective

Derek Fowler

Independent Safety Engineering Consultant, Bracknell, UK

Abstract

This article is the fourth in a series of papers that address the potential application of international generic functional-safety standard IEC 61508 to the transport sector, with a focus on the crucial requirements/specification phases of the system safety lifecycle in Part 1 of the Standard (IEC 61508-1). During the peer reviews of the three predecessor articles, questions were raised about the degree and extent to which some European Air Traffic Management and Rail safety-assessment standards and practices actually complied with the fundamental principles of IEC 61508-1; however, at the time, it had not been the intention of any of those three articles to address such questions, and it was left to readers to reflect on the matter. This article now returns to the subject of safety compliance, but in the context of Safety Assurance, and starts by developing a generic safety-assurance framework, based on classical systems-engineering practices and rigorous requirements-engineering principles, against which the IEC 61508-1 safety-assessment processes, and their equivalents in European Air Traffic Management and Rail safety standards, are judged in terms of their potential contribution to the safety case for typical safety-related transport applications.

1 Introduction

1.1 Background

In the first of four related articles, Fowler (2022) proposed ‘a way of thinking’ about safety assessment of the various safety-related systems deployed in the Transport sector in general — but especially in commercial-aviation and rail applications — that was based closely on the application of the key principles and safety lifecycle set out in Part 1 of IEC 61508 (hereinafter referred to as IEC 61508-1).

A follow-up article took the example of an innovative Air Traffic Management (ATM) operational concept and used it to outline how an IEC 61508-1 approach to safety assessment could be applied effectively to the ATM sector, and what the results thereof might look like; that approach started with the concept of the movement of aircraft in the airspace being (what IEC 61508-4 defines as) the Equipment Under Control (EUC) (Fowler and Fota 2023).

A third article then took a similar approach but with the example of a new, moving-block Communications-based Train Control (CBTC) system for a hypothetical metropolitan

railway; this was used to outline how an IEC 61508-1 approach to safety assessment could be applied effectively to the Railway sector in general, and what the results thereof might look like; that approach started with the concept of the movement of trains around the railway being the EUC (Fowler and Graebner 2023).

This, the fourth and final article in the series, takes a somewhat different, argument-based, assurance perspective motivated in part by:

- The findings of Fowler (2015), which suggested that neither the CENELEC set of railway standards (in particular the 1999 version of EN 50126 and the 2003 version of EN 50129), nor most national ATM safety-assessment standards¹, applicable at **that** time, complied with even the most basic principle of the IEC 61508-1 safety lifecycle;
- An understanding that significant changes in ATM and Rail safety standards have been made since 2015; and
- The following, rather bold, statement, in the 2017 version of EN 50126-1:

“EN 50126 forms part of the railway sector specific application of IEC 61508. Meeting the requirements in this European Standard, together with the requirements of other suitable [unspecified] standards, is sufficient to ensure that additional compliance to IEC 61508 does not need to be demonstrated”.

So, what does compliance to IEC 61508 mean in practice and, in any event, what would be the value of the IEC 61508-1 lifecycle processes to system safety in the transport sector? Clearly, addressing these two rhetorical questions would require a rigorous and independent suitable benchmark on which to base informed decisions. It was decided to develop, and to present herein, an Argument-based Safety Assurance Framework for that purpose.

1.2 Historical Context

1.2.1 Hard Lessons

Over the period 1970 to 2002, there were nearly 40 major fatal accidents reported worldwide, involving more than 8,000 immediate fatalities, and across three industry sectors, as follows²:

- Aviation: at least 17 major accidents, involving more than 3,700 fatalities;
- Chemical and Process Industries: at least 12 major accidents, involving more than 4,000 fatalities;
- Rail: at least 8 major accidents, involving more than 550 fatalities.

Individually, these accidents shaped modern approaches to systems engineering, human factors, software assurance, safety culture, and risk management. However, the concept of a Safety Management System (SMS) did not emerge all at once; rather, it evolved gradually through the 1970s, 1980s and 1990s, as industries moved away from viewing accidents as purely technical failures and toward also addressing management, organisational and human factors. Some of the earliest recognisable SMS-like approaches appeared in response to

¹ In retrospect, that comment would have been better confined to European countries, although the author was not aware of any countries that would have had better safety assessment standards at that time!

² These are intended as indicative figures only, and were obtained from an AI-assisted survey. They exclude harmful outcomes caused by deliberate acts of terror, etc.

accidents in high-risk industries such as nuclear power, aviation, chemical processing, offshore oil and gas, and railways. For example:

- **Tenerife Airport Disaster** (1977), in which two aircraft colliding during take-off, brought focus to the need for improved crew communication and cockpit resource management;
- **Three-Mile Island Nuclear Plant Accident** (1979), in which cooling-system failure caused a partial reactor meltdown, led to enhanced operator training, human factors engineering, and the creation of the US Institute of Nuclear Power Operations to promote safety culture;
- **American Airlines Flight 191 Crash** (1979), which remains one of the most studied aviation disasters in history, because it demonstrated how maintenance-related damage can trigger catastrophic system failures. It is frequently cited in aviation safety training and accident investigations as an example of the importance of maintenance standards, engineering redundancy, and regulatory oversight;
- **Bhopal (India) Chemical Plant Disaster** (1984), which involved the accidental release of 40 tons of methyl-isocyanate gas, leading to around 3,000 immediate deaths, tens of thousands of deaths in the longer term, and up to half a million people with permanent disabilities. The accident triggered a multi-decade global evolution in how governments and corporations manage industrial hazards, covering the period 1985 to 2024, and involving the relevant authorities of Canada, India, the EU, the UK, and the USA (Willey et al. 2005) (Cogent Skills 2026) (Mannan et al. 2005).
- The report on the **Challenger Space Shuttle Disaster** (1988), in which failure of an O-ring seal in the right solid rocket booster at liftoff allowed hot gases to escape and damage the external fuel tank, leading to the disintegration of the orbiter, cited flawed engineering decisions, inadequate risk communication, and managerial pressures to maintain launch schedules as key causal factors, and led to sweeping changes in NASA's safety culture, management, and design protocols;
- **Piper Alpha Oil Platform Disaster** (1988), in which a maintenance error caused gas-condensate leak and ignition, and led to triggering multiple explosions, and fire spreading rapidly as adjacent gas pipelines ruptured, engulfing the platform in flames within minutes — of 226 personnel on board, only 61 survived. The Cullen Inquiry report into the disaster (Cullen 1990) fundamentally revolutionised offshore safety by recommending that operators be legally required to produce a comprehensive Safety Case as a means of assuring itself that its operations are [acceptably] safe. In this way, it shifted the regulatory regime from a prescriptive checklist of rules, to a "goal-setting" approach where operators must actively identify risks and demonstrate how they manage them; and
- **Charkhi Dadri Mid-Air Collision** (1996), in which a Saudia Boeing 747 and a Kazakhstan Airlines Ilyushin Il-76 collided at 14,000 feet over India due to language barriers, poor English comprehension by the Kazakh crew, and antiquated ground radars. Following this disaster, the International Civil Aviation Organization (ICAO) made the Airborne Collision Avoidance System (ACAS) mandatory for all major commercial aircraft worldwide. ACAS allows aircraft to communicate directly with each other in mid-air and issue automated "climb" or "descend" commands to pilots, with the intention of completely bypassing airborne and ground-based human error — but see Sub-section 4.4.2 below!

All of the above, and others, provide important context for this article, in terms of an evolution of safety management during the last three decades of the twentieth century, which moved gradually **from**:

- Investigating accidents **after** they occur to determine their causes and prevent similar incidents from happening again; **to**
- Identifying hazards **before** accidents happen to proactively recognize potential sources of harm and implement preventive measures; **and to**
- Managing risk **continually** throughout a system's lifecycle by regularly assessing, monitoring, and controlling risks from design and development through operation, maintenance, and decommissioning.

1.2.2 A New Generic Approach

In 1985, the International Electrotechnical Commission (IEC) initiated an evaluation of the viability of a generic standard for Programmable Electronic Systems (PES) used in safety applications, and in 1995 distributed a draft, known as "IEC 1508", which introduced early concepts of Safety Integrity Levels (SIL) and lifecycle frameworks to specific manufacturing sectors. In the period 1998 to 2000, the seven parts of the first full edition of what became IEC 61508, "*Functional safety of electrical/ electronic/programmable electronic [E/E/PE] safety related systems*", were released and became probably the most widely accepted, international generic standard on functional safety. Although its roots can be traced to process industries, the intention behind the Standard had always been, and still is, to provide a solid, comprehensive basis for adaptation to a very wide range of industry sectors.

IEC 61508 is about Functional Safety, which it defines as "*part of the overall safety, relating to the EUC and the EUC Control System, that depends on the correct functioning of the safety-related systems and other risk-reduction measures*" — and which leads to its very specific approach to systems safety assessment. This concept was described in some detail in Fowler (2022) but, because it is not always understood, yet has a major influence on the Safety Assurance structure developed herein, the key points are worth restating here, as follows:

- IEC 61508 is founded on the notion of an EUC that is defined as "*equipment, machinery, apparatus or plant used for manufacturing, process, transportation, medical or other activities*", and which is inherently hazardous to human life and/or the environment;
- The EUC might have an associated, separately identifiable, EUC Control System that "*... responds to input signals from the [EUC] and/or from an operator, and generates output signals causing the EUC to operate in the desired manner*";
- Safety-related Systems (SRSs) are developed in order to implement the Safety Functions that are necessary to mitigate the pre-existing³, EUC Hazards, thereby achieving, or maintaining, an "*acceptably safe state for the EUC*";
- Safety Integrity is "*the probability of an SRS satisfactorily performing the specified Safety Functions, under all the stated conditions, within a stated period of time, so as to achieve, and maintain, an acceptably safe state for the EUC*". As emphasised throughout this article, it is NOT simply the likelihood of failure of an SRS.

Note: A more comprehensive introduction to these crucial principles, which underpin IEC 61508, can be found in Redmill's (1999) introduction to the standard; it is as valid and important today as it was at the time!

³ The hazards that are inherent in the EUC are known as "pre-existing" herein since they pre-date the existence of the SRS(s), and to distinguish them from any hazards that might be introduced by failure of said SRS(s).

1.2.3 The Leveson Contribution

During the period of this IEC 61508 activity, and beyond, Professor Nancy Leveson of MIT had been working in what was then a new area of software safety and reliability, including a major, detailed study into a variety of spacecraft and aircraft accidents that in some way involved software. A helpful overview of this seminal work, and its conclusions, was presented at the 19th International System Safety Conference, in 2001 (Leveson 2001), and, in addressing the rhetorical question as to how many of the accidents were caused by software reliability problems, it referred to:

“... confusion between software reliability and safety: most accidents [studied] involve software that is doing exactly what it was designed to do, but the designers misunderstood what behavior was required and would be safe, i.e. it reliably performs the wrong function[s]”.

More generally she concludes that:

“I have noticed ... that, over the years, system safety engineering has increasingly drifted toward using reliability engineering techniques and away from classic system safety approaches and has, in particular, adopted this approach for software. This trend may simply be a result of lack of knowledge about software, or it may reflect a lack of appropriate tools to assist in applying system safety approaches to software. A further influence may be that computer science has always been concerned with computer [hardware] reliability and has focused almost exclusively on this quality. Only recently has safety become an issue. Therefore, almost all existing software engineering techniques focus on software reliability, i.e. assuring that the software correctly or reliably satisfies the specified [functional] requirements (which [themselves] may be incomplete, incorrect, or unsafe).”

In other words, her findings very much articulate the problems that the early contributors to IEC 61508 set out to address.

It is very disappointing, therefore, that nearly 20 years later Professor Leveson found it necessary to describe, in an Association of Computing Machine (ACM) opinion piece (Leveson, 2020), four major misconceptions which suggest that:

“... confusion and misinformation [still] abound about software and safety Too many incorrect beliefs are being promoted, which are inhibiting progress and, in some cases, unnecessarily costing lives”.

Most importantly, she concludes that:

“... the most effective approach to dealing with safety of computer-controlled systems is to focus on creating the safety-related requirements. System and software requirements development are necessarily a system engineering problem, not a software engineering problem. The solution is definitely not in building a software architecture (design) and generating the requirements later, as has been surprisingly suggested by some computer scientists”!

Again, that is fully in tune with the provisions of IEC 61508-1, and is what this article, (in particular, Section 4 below) is all about.

1.2.4 Developments in European Air Traffic Management Safety Management

From the early 1990s, the UK National Air Traffic Services (NATS) pioneered the early development of a formal SMS, having been mandated by the UK government to operate independently from its regulatory body in the UK Civil Aviation Authority (CAA). That SMS contained all the recognised elements of an SMS; however, like many equivalent system safety-assessment processes at the time, and quite unlike what emerged subsequently in IEC 61508, followed traditional reliability-engineering methods and was based on an assumption that system failures would occur in a random fashion. At the time, such an approach would have been reasonable, since Air Traffic Management (ATM) functional safety rested primarily on the skills and experience of human Air Traffic Controllers, and the hardware tools on which they relied contained little or no software safety functions.

In the late 1990s the UK CAA, being increasingly concerned about the new generation of software-intensive safety-related systems being introduced into ATM, commissioned the development of a set of comprehensive safety objectives, known as “SW01”, and first published them formally in December 2002, in what has now become document CAP⁴ 670 (UK CAA 2019). SW01 was, and still is, widely regarded as seminal work, but with one stated significant deficiency:

“This document assumes that software safety requirements have been derived from a full risk and safety analysis of the system. This will have established the overall safety requirements that have been refined and allocated in the design to software.”

Whereas it rightly claims that *“This is a commonplace system safety process and is described in standards and guidelines such as IEC 61508 Part 1 ...”*, some system-safety practitioners might find it disappointing that no equivalent, system-level safety objectives have been published.

Meanwhile, on a wider, EUROCONTROL⁵ front, parallel developments were taking place. Until the mid-1990s, it had been recognised that safety was an implicit but key feature of ATM operations. Technical systems were functionally quite basic, and the primary role of ATM — that is, reducing the risks that are inherent in aviation, especially the risk of mid-air collisions, or high-energy collisions between aircraft on the ground — was vested in highly skilled, well-motivated human operators (controllers) working to well-defined procedures.

With the increased use of technology, and the introduction of formal safety management systems in ATM during the second half of the 1990s, EUROCONTROL started on the development of its very-detailed “Safety Assessment Methodology” (SAM) (EUROCONTROL 2006a).

The original intention was to base the SAM on the well-established commercial-aircraft-certification procedures, first published in 1996, SAE ARP 4754A and SAE ARP 4761.

SAE ARP 4754A details the engineering processes used to develop civil aircraft and systems. It focuses on:

- bridging high-level aircraft functions down into specific system requirements.
- ensuring the right system is being built and that the system is built correctly.

⁴ Civil Aviation Publication

⁵ The European Organisation for the Safety of Air Navigation

- defining Development Assurance Levels (DALs) to manage the rigour of hardware and software design.

SAE ARP4761 provides the methods and guidelines to execute the safety assessments mandated by SAE ARP4754A. It works in lockstep with the development process to:

- classify the severity of potential failure conditions using Functional Hazard Assessments (FHA).
- analyse system architecture using techniques like Fault Tree Analysis (FTA) and Failure Modes and Effects Analysis (FMEA) to prove safety requirements are met.
- identify common-cause or cascading failures that could bypass independent systems.

As it turned out, very little of SAE ARP 4754A was adopted for the SAM and so what was promulgated by EUROCONTROL was an entirely failure-based approach that utilised traditional reliability-engineering techniques — contrary to the principles underpinning IEC 61508 and those expounded by Nancy Leveson at the time, and since.

It was not until about 2008, in the face of the enormous challenges presented by the European Commission’s Single European Sky ATM Research & Development (SESAR) programme, that EUROCONTROL, with others, started to develop a far more rigorous and capable system safety assessment method (SESAR 2018a) with guidance for its application (SESAR 2018b), which was much more closely aligned with the principles behind SAE ARP 4754A and IEC 61508-1 — see later in Sections 5 and 6 below.

1.2.5 Developments in European Rail Safety Management

During the 1990s, the privatisation and cross-border integration of railways created a critical need for universal safety benchmarks. In response, the European Committee for Electrotechnical Standardization (CENELEC) drafted a sector-specific adaptation of the generic functional safety standard IEC 61508. The original standard was approved as EN 50126:1999, “*Specification and Demonstration of Reliability, Availability, Maintainability and Safety (RAMS)*”, and introduced the well-known 14-phase System (V-model) life cycle to the rail industry.

This was followed by the addition of two related standards:

- EN 50128, “*Software for Railway Control and Protection Systems*”, which defines processes and technical requirements for developing, verifying, validating, and maintaining software used in railway control and protection systems; this was first published formally in 2001; and
- EN 50129, “*Railway Applications – Communication, Signalling and Processing Systems – Safety Related Electronic Systems for Signalling*”, which defines the requirements for the acceptance and approval of safety-related electronic systems used in railway signalling, and was first published formally in 2003⁶.

The problem with EN 50126, in particular, was that it was based on (or could be interpreted as being based on) what Nancy Leveson would describe as “*reliability-engineering techniques*” and, as suggested by Fowler (2015), did not comply with even the most basic principles of IEC 61508-1.

⁶ But published earlier (1998) as a European “pre-standard” (ENV 50129)

Since then, there have been several major updates to the above CENELEC standards:

- **EN 50126**, in its latest, 2017, version has been split into two parts: EN 50126-1, which is the administrative and managerial backbone of the lifecycle. It governs the entire timeline and covers all four RAMS pillars; and EN 50126-2, which is a highly specialized, technical engineering manual dedicated exclusively to Safety;
- The original 2001 edition of **EN 50128** was a straightforward railway adaptation of IEC 61508-3. It then gradually shifted from being primarily a software development standard to being a broader software assurance standard; with more emphasis on organisational competence, management processes, tool confidence, verification independence, and demonstrating confidence in the overall software engineering process, rather than focusing predominantly on implementation techniques⁷;
- The 2003 edition of **EN 50129** established the basic framework for producing a safety case for safety-related electronic signalling systems. In 2018, a major revision aligned it with the EN 50126:2017 lifecycle, updated the safety case requirements, and incorporated guidance from CENELEC documents CLC/TR 50451 and CLC/TR 50506. What is particularly relevant to this article are the safety case aspects, which could be summarised as shifting from "*produce these documents*" to "*demonstrate a convincing safety argument*".

This then brings us back to the question of whether the claim that “*EN 50126 ... together with the requirements of other suitable [unspecified] standards, is sufficient to ensure that additional compliance to IEC 61508 does not need to be demonstrated*” was justified when it was made by CENELEC in 2017.

In the abstract world of IEC 61508, the EUC is the *problem* since, by definition⁸, it is inherently hazardous. The SRS, on the other hand, is the *solution* since, by definition, it provides the Safety Functions that mitigate the hazards that exist within the domain of the EUC. The first step in trying to apply IEC 61508 to a specific sector is to determine a suitable practical interpretation of the “EUC” — in the worked example of Fowler and Graebner (2023), we saw that “*the movement of rail traffic*” was the most suitable interpretation for the rail sector. Similarly, and bearing in mind that the rail example given therein is based on a major upgrade to existing railway operations, it was considered appropriate for the legacy railway infrastructure to be interpreted as being the EUC Control System⁹. By the same token, the SRSs were constituent parts of the new automatic train control system that was being developed for the hypothetical railway.

In the practical world of the above CENELEC standards, things seem (to the author of this article at least) to be less clear. Those standards do not use the term EUC explicitly; rather they refer to, the much more flexible notion of the *System under Consideration* (SuC). Furthermore, whereas the concept of a hazard as a “*condition that could lead to an accident*” (as per (EU) No 402/2013) is on the face of it roughly equivalent to the IEC 61508 definition of “*a potential source of harm*”, only the CENELEC usage, in order to comply (EU) No

⁷ In 2023, the 2020 edition was superseded by EN 50716 in what might be viewed as a refinement and modernisation rather than a complete rethink of railway software assurance.

⁸ In IEC 61508-4

⁹ Of course, other interpretations might be equally valid provided that, if properties of part, or all, of the EUC Control System had to be relied upon in the achievement of an acceptably safe state for the EUC, then those aspects of the EUC Control System would have to be treated as being safety-related, in accordance with the Standard.

402/2013 makes the following strong link between hazards and “the system” — presumed to be the SuC¹⁰:

“The risk assessment process starts with the system definition (which can use information from the preliminary system definition). This provides the key details of the system that is being changed - its purpose, functions, interfaces and the existing safety measures that apply to it. In most cases, the hazards which need to be analysed will exist at the boundary of the system with its environment.”

Herein lies a dilemma. If we were to define the SuC as being the whole railway — i.e. equivalent to the EUC — then the hazard-identification process would (or should) start with the hazards that are inherent in railway operations, as per an IEC 61508-1 approach. However, if the SuC were to be defined as being, say, a new automatic train control system that was being introduced to the railway, what “hazards” would we then be identifying?

It seems that the CENELEC standards are aimed more at a change to a railway, rather than at the whole railway operation, whereas IEC 61508 is clearly aimed at the whole operation in whatever sector to which it is to be applied. There *should* be, however, no contradiction here because, for example, in order to specify a complete set of safety requirements for a new automatic train control system, one should *first*:

- have a complete understanding of the EUC, the EUC Control System and the Operational Environment and the interactions between them and the SuC (i.e. the new system);
- know the parameters of all of the pre-existing operational (or EUC) hazards that the new system would have to mitigate;
- assess all of the impacts that the change (i.e. the introduction of the new system) would have on the existing operating rules (e.g. signalling principles) for the railway, captured in what IEC 61508 calls *Overall Safety Functions / Overall Safety Requirements*.

However, there is a real danger here that too literal an interpretation of the term SuC would lead to a mindset that the hazards to be assessed would be those caused by the SuC — in effect, the problem Leveson (2001) consistently warned against and reaffirmed more recently in Leveson (2020). As we will see in, *inter alia*, the case history outlined in section 4.4.3 below, we could easily end up with a set of system safety requirements that specified only a tolerable failure rate for the system, and not also the functionality that the system would need in order to mitigate the pre-existing, railway-level, operational hazards.

Of course, this is all about assurance activities, and perhaps, therefore, the question should be, not whether the processes in EN 50126 *et alia* comply with the corresponding processes of IEC 61508, but whether, by careful interpretation of the system under consideration and adherence to the standards themselves, the results would then produce assurance evidence sufficient to stand up to the demands of a rigorous safety argument. It follows, therefore, that this question would best be addressed.

1.3 Scope

Like its predecessors, this article focuses mainly on the crucial, early phases of the IEC 61508-1 safety lifecycle — i.e. on the *Specification* of Overall Safety Requirements (Phases 1 to 4), and the high-level *Design* of the System (Phases 5, 9 and 11). This is because most of the key principles underpinning IEC 61508 — i.e. those universal principles set out in

¹⁰ The ORR (2024) guidance document effectively confirms the application of (EU) No 402/2013 to the UK, post-“Brexit”.

Part 1 of the Standard, which govern the determination of the required risk-reducing properties of Safety-related Systems — take effect during these earlier phases, whereas the subsequent implementation and operating phases are less specific to the Standard.

Indeed, the truth of this assertion is borne out by Leveson (2020), as quoted in Sub-section 1.2.3 above and, in Sub-section 4.4 below, by the three past examples of where the ignorance of the above principles appears to have led to disastrous consequences!

1.4 Aims and Objectives

The aim of this paper is to develop a generic Safety Assurance structure, against which the efficacy of transport system safety-assessment standards could be judged.

This aim is addressed through the following objectives:

1. the development of an Argument-based Safety Assurance framework, compatible with the safety lifecycle of IEC 61508-1, but based predominantly on well-founded requirements-determination and systems-engineering practices, and focussed on safety-requirements specification and high-level design;
2. assessment of the strengths and shortcomings of IEC 61508-1 processes in the development of evidence to support the arguments for safety-requirements Specification and high-level Design; and
3. assessment of the efficacy of the illustrative products of the application of IEC 61508-1 to the ATM sector (Fowler and Fota 2023) and Rail sector (Fowler and Graebner 2023), in providing evidence to support the arguments for system safety-requirements Specification and high-level Design.

2 Safety Assurance Principles and Rationale

2.1 Safety Assurance Definition

Although the term “Safety Assurance” is widely used across many safety-related industry sectors, there does not appear to be a single, internationally accepted definition of the term. A recent AI-assisted search of its usage suggested the following, which is judged to serve the purposes of this article very well:

Safety assurance is the planned and systematic set of arguments, activities and evidence, which demonstrate, with justified confidence, that a product, service, organisation, or functional system is acceptably safe, in a given context.

2.2 The Need for an Argument-based Safety Assurance Approach

In Annex 5 to its guidance on the introduction of AI into organisations (ICO 2022), the UK Information Commissioner’s Office draws an interesting comparison between assurance-case approaches, “... *as have historically arisen in safety-critical domains ...*”, and which it strongly recommends for the introduction of AI into organisations, and what it calls “*the older, more reactive and prescriptive methods that are now increasingly challenged by the complex and rapidly evolving character of emerging AI technologies*”.

It goes on to explain that “*An assurance case is a set of structured claims, arguments, and evidence which gives confidence that an AI system will possess the particular qualities or properties that need to be assured...*” and that this “*...would involve providing an argument, supported by evidence, that a system possesses the properties that will allow it to function safely, securely, reliably, etc., given the challenges of its operational context*”.

Whereas the “*older prescriptive approach stressed the application of one-size-fits-all general standards that specified how systems should be built and often treated governance as a retrospective check-box exercise*”, it suggest that “*argument-based assurance takes a different tack, ...from the inception of an AI project and plays an active role at all stages of the design and use lifecycle. It begins with high-level normative goals derived from context-anchored impact and risk-based assessment for each specific AI application and then sets out structured arguments demonstrating:*

- *how such normative requirements address the impacts and risks associated with the system’s use in its specified operating environment;*
- *how activities undertaken across the design and deployment workflow assure the properties of the system that are needed for the realisation of these goals; and*
- *how appropriate monitoring and re-assessment measures have been set up to ensure the effectiveness of the implemented controls*”.

It notes that “*... argument-based assurance cases have historically arisen in safety-critical domains ... because it is a reasonable way to structure and document an anticipatory, goal-based, and procedural approach to innovation governance, ... in contrast to the older, more reactive and prescriptive methods that are now increasingly challenged by the complex and rapidly evolving character of emerging AI technologies*”.

It is for very similar reasons that EUROCONTROL had previously used argument-based safety assurance for the introduction of Reduced Vertical Separation Minima for aircraft across most of European airspace in 2002 (Fowler and Tiemeyer 2002), as has the European Commission’s Single European Sky ATM Research & Development programme since its inception (Fowler et al. 2009).

It is also the reason for selecting such an approach for this article, for assuring the safety of other highly complex systems / operations!

2.3 Safety Assurance Model

Figure 1 is a Safety Assurance Model, which has been adapted from Fowler *et alia* (2009), and fits the above definition very well.

The green box represents the whole of the Argument structure, with all of its contextual annotations, each and every strand of which has to be provided with supporting Evidence to show that it is valid (i.e. true). An example notation for such a structure is described in Section 3 below.

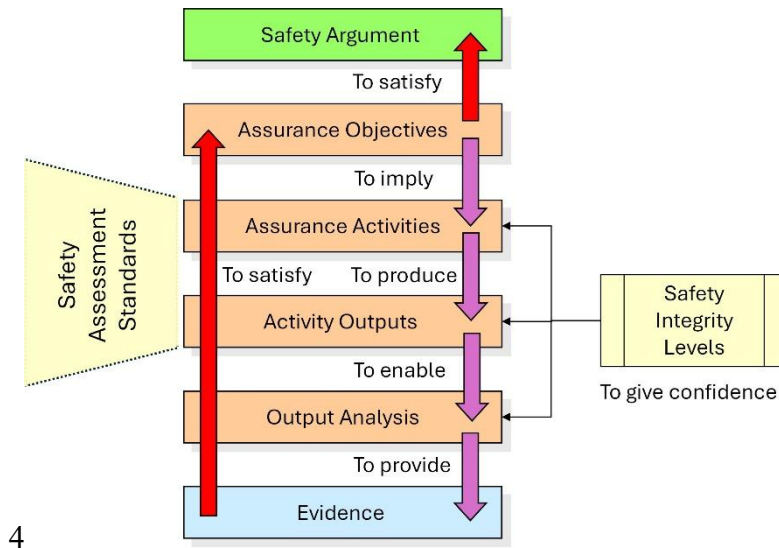


Figure 1~ Safety Assurance Model

Evidence is any artefact or item of information that supports the truth of an Argument and that is shown to be relevant, credible, sufficient, traceable, and verifiable.

Assurance Objectives state what the Evidence has to demonstrate in order to satisfy the associated Argument statement¹¹ — in effect, therefore, they are actually the lowest level of decomposition of the Argument. The Assurance Objectives specified in Sections 5 and 6 below were adapted in part from CAP670 SW01 (UK CAA 2019), which sets out a very comprehensive set of “*Regulatory Objectives for Software Safety Assurance in ATS*”¹².

The Assurance Objectives imply the *Assurance Activities*, i.e. the safety-assessment processes, that are necessary and sufficient to satisfy each Assurance Objective. The *Activity Outputs*, i.e. the results of completing the Assurance Activities, form the basis of the *Evidence*, subject to the addition of the results of *Activity Output Analysis* that show (with the appropriate degree of rigour) how the Evidence satisfies the related *Assurance Objective*.

Figure 1 also suggests that safety-assessment standards, such as IEC 61508, usually focus on Assurance Activities and Assurance Outputs, with some attention also to Output Analysis. IEC 61508-1 does *not*, however, address what are defined above as Assurance Objectives — its use of the term Objective, is very different in that, in most cases, they are phrased as things that have to be done, i.e. Assurance Activities, *rather* than things that have to be demonstrated.

A major positive contribution of IEC 61508 is its use of Safety Integrity Levels (SILs)¹³ to define, *inter alia*, the *rigour* demanded of the system-development processes and the safety-assessment processes: Assurance Activities, Outputs and Analysis.

¹¹ Strictly speaking, the way they are usually worded does not preclude a further decomposition of the Argument should that prove necessary.

¹² Air Traffic Services

¹³ Sometimes referred to as Safety Assurance Levels (SALs)

3 Developing an Argument-based Safety Assurance Structure

3.1 Structure and Notation

Safety assessments are often done within the context of a Safety Case which might, like a legal case, comprise two main elements:

- a set of arguments, i.e. statements which claim that something is true (or false); and
- supporting evidence to show that each argument is actually true.

Safety arguments are normally set out hierarchically. In this case, an *adapted form* of Goal-structuring Notation (GSN) has been chosen for the purpose, as shown in Figure 2. GSN is simply a graphical representation of an argument / evidence structure and usually starts with the top-level *claim* (Arg 0) that something is (or will be) tolerably (or acceptably¹⁴) safe; this is then decomposed such that it is true if, and only if, the next-level *argument* statements (in this case Arg 1 to 4) are themselves *all* true. The *strategy* text (St001) should explain the rationale for that decomposition.

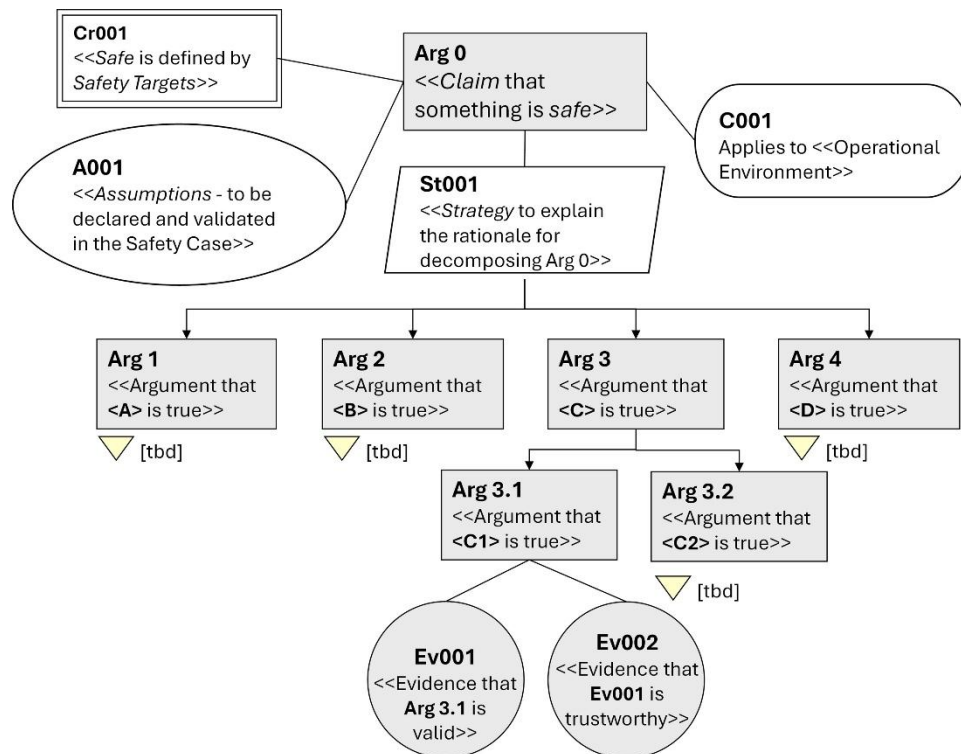


Figure 2 ~ A Generic High-level Safety Argument

The claim (Arg 0) is supported by vital contextual information, as follows:

- what is meant by *acceptably safe* is defined by means of *safety targets* (see *constraint* Cr001);
- the *context* (C001) for the claim must include a description of the operational environment for which the claim is being made — it will be seen in Sub-section 4.1 how critical this is to the validity of the claim;

¹⁴ i.e. Tolerably safe plus ALARP (As Low as Reasonably Practicable) (UKMoD 2016).

- *assumptions* (e.g. A001) are sometimes used for material on which a claim or argument depends but the validity of which might be in some doubt; it is vital that such assumptions are shown to be reasonable and that they are logged and tracked through to the end of the safety lifecycle.

The arguments would then be further sub-divided until a level is reached at which documented evidence, of a manageable size, could be produced to show that the related argument is valid.

Figure 2 identifies two main types of Evidence, in support of Arg 3.1:

- Evidence that shows directly that a particular objective has been achieved — in this case, that Arg 3.1 has been satisfied — and is referred to as *Direct* evidence (Ev001);
- Evidence that shows that the Direct evidence is trustworthy, i.e. that it can be relied upon. This is referred to as *Backing* evidence (Ev002).

Appendix A hereto presents a comprehensive set of principles for developing safety-assurance structures and supporting evidence, which have been adapted from EUROCONTROL (2006b), and are recommended to be followed in the development of assurance structures (and have been followed herein!).

3.2 Transport Illustration of a Top-level, Lifecycle Safety Assurance Structure

Figure 3 is an illustration of a top-level safety-argument structure, and has been worded such that it could be easily adapted to most safety-related transportation upgrades.

It should be noted that the top-level Claim is deliberately made for an acceptable level of safety of the operations enabled by the System, not for the System itself — the necessity for this distinction should become clear in the detailed discussion in Section 4.

The question then is how to ensure that the next level of the safety argument would be complete and rigorous — for this, an approach is proposed that follows a reasonably conventional, 5-stage systems-engineering development process, and starts with a high-level safety argument for each stage, as follows (Fowler et al 2009).

Arg 1 — argument that the System (or system of systems) to be introduced has been specified to be safe, i.e. the overall safety requirements (or *Specification*; see Section 4) meet the appropriate *Safety Targets* for the Operational Environment completely and correctly.

Arg 2 — argument that the high-level *Design* of the System satisfies its *Specification*, correctly and completely.

Arg 3 — argument that the *Implementation* of the *Design*, in the built-and-tested System, satisfies the *Design* requirements correctly and completely.

Arg 4 — argument that the *Transition* from current signalling system to the new System will be safe, i.e. the known risks during this process have been reduced to a level that is As Low as Reasonably Practicable (UKMoD 2016).

Arg 5 — argument that the System will be shown to continue to operate safely throughout its Operational Service.

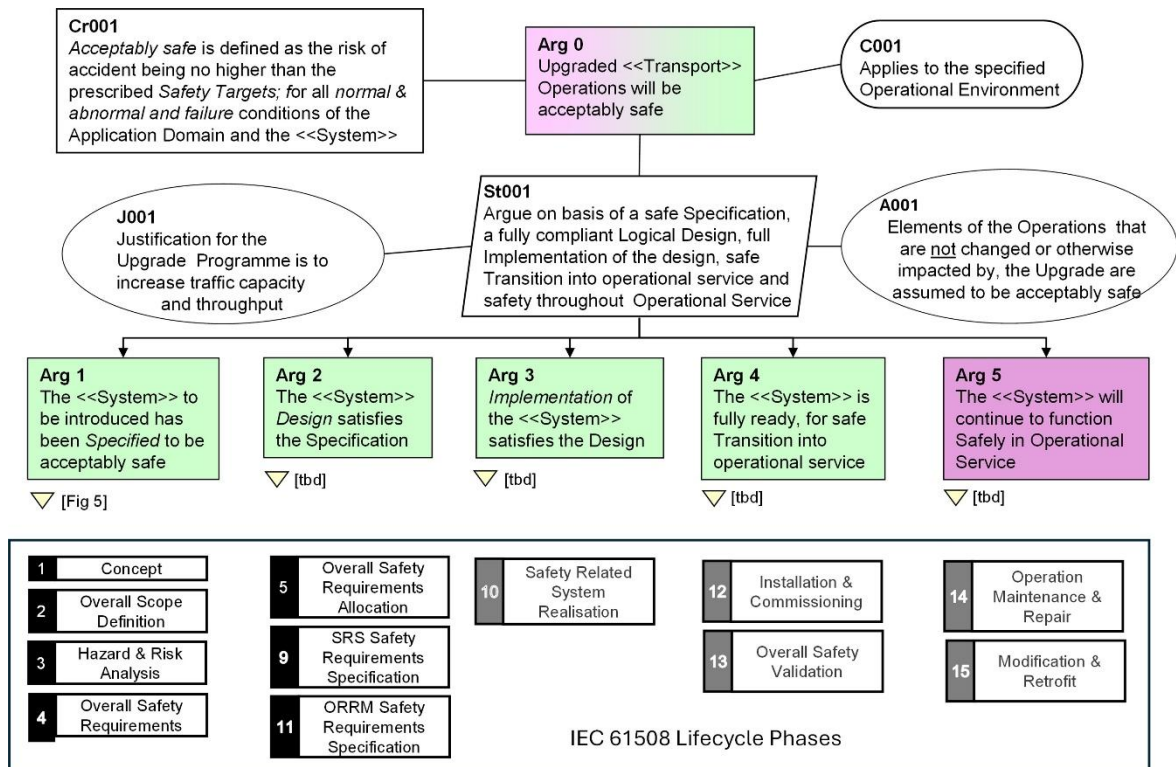


Figure 3 ~ Generic Top-level Safety Argument for a Transport Application

For ease of reference only, the general relationship with IEC 61508-1 safety-lifecycle phases (except for three planning phases and Phase 16-*Decommissioning*) is shown at the foot of the argument diagram, the phases that are outside the scope of this article being shown in grey. It is important to note, however, that the Safety Argument was *not* derived from those IEC 61508-1 phases, and its purpose is to provide a generic model for assessing the ability of transportation safety-engineering standards (and of IEC 61508-1 itself!) to provide incontrovertible direct and backing evidence of adequate system functional-safety properties.

In line with the scope of the three previous articles, the further decomposition of the argument, discussed in Sections 5 and 6 below, focuses entirely on Arg 1 and Arg 2, especially the former. Since those two arguments are based heavily on sound systems-engineering practices, they are described in Section 4 below, along with some examples of the pitfalls of not following them.

4 Systems Safety Engineering — Good Practice and Pitfalls

This section describes good practices in systems engineering on which the subsequent decomposition of Arguments 1 and 2 are based, and presents some real-life examples of the perils of ignoring them.

4.1 Requirements Engineering Concepts

In his seminal work on *The World and the Machine* (Jackson 1995), Dr Michael A Jackson set out crucial scientific principles regarding requirements capture. He summed up the perceived requirements-capture problem, as follows:

“We are concerned both with the world, in which the machine serves a useful purpose, and with the machine itself. The competing demands and attractions of these two concerns must be appropriately balanced. Failure to balance them harms our work”.

The concepts underlying his Problems Frame Approach were adopted by the then Praxis Critical Systems Ltd.¹⁵ in the development of its REVEAL[®] Requirements Engineering method, including the simple, but rigorous, Requirements Engineering model shown in Figure 4 (Hammond et al. 2001).

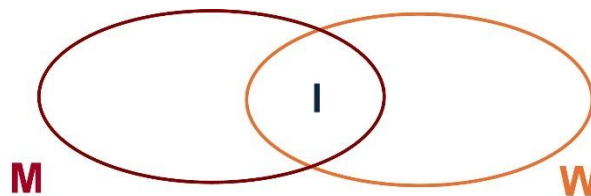


Figure 4 ~ The World and the Machine

Essentially, *Machines* (M) are built in order to bring about some improvement to the *World* (W). In these terms, a Machine can be any socio-technical artefact that is to be constructed, and need not be physical, e.g. it could be a process.

Whenever a Machine is introduced to a World, it interacts with that World through a shared *Interface* (I), and items that might occur in this interface are those phenomena that are shared between the World and the Machine; they are *shared* because they are in some sense visible to both the World and the Machine, i.e. they represent the Machine’s view of the World, and vice-versa. It is through these shared phenomena that the Machine seeks to influence the behaviour of the World. Other shared phenomena could be controlled by the World and responded to by the Machine.

The following definitions are based on those provided by Jackson (1995):

- *Requirements* (**R**) are a collection of statements about phenomena in the World that we want the Machine to *help* make true — the implication is that it is not solely the Machine that brings about the Requirements;
- a *Specification* (**S**) is a collection of statements that describes a Machine’s required external behaviour. A Specification statement refers only to shared phenomena in the Interface, and can only constrain those shared phenomena that the Machine itself could control;
- the *Application Domain* covers those aspects of the World that are relevant to the improvement that we wish the Machine to achieve, and its properties — that we know (or assume) to be true — are termed *Domain Knowledge* (**D**).

¹⁵ Later Altran UK Ltd and Capgemini.

This recognition of the rôle of Domain Knowledge in the Requirements Engineering process leads Jackson to suggest the following relationship, which must hold if a Machine is to be introduced to a World to bring about some Requirements:

$$\mathcal{D}, \mathcal{S} \vdash \mathcal{R}$$

Hammond *et alia* (2001) call this a “*satisfaction argument*”, which should be read as follows:

“Using the relevant properties of the Application Domain ($\mathcal{D}$), when combined with the Specification ($\mathcal{S}$) of the behaviour of the Machine to be constructed, it is possible to show ($\vdash$) that the Requirements ($\mathcal{R}$) will hold”,

and is the foundation of Arg 1, as discussed in Section 5 below.

4.2 Rich Traceability

Of course, we must demonstrate that the $\mathcal{D}, \mathcal{S} \vdash \mathcal{R}$ does actually hold true in practice and, for this, Hammond *et alia* (2001) recognise that a traditional, basic requirements-traceability approach would not be adequate because it would simply link:

“... items from one artefact to the corresponding items from a previous artefact, or to the source of the items. For example, a specification statement may be linked to all the requirements that it helps to satisfy. However, there is no explanation or justification of the links and so their semantics is often unclear. Typically, all that a link between two items can suggest is that changes in one may require changes to the other”.

Instead, Hammond proposes a “Rich Traceability” approach, “... which focuses on **how** an item is satisfied, by enhancing the use of simple links with additional structure and justification”.

In essence, *rich traceability* is a more comprehensive, multidimensional way of capturing, linking, and analysing traceability information across the lifecycle of a product, system, or process, by:

- capturing different types of artifacts (requirements, design, code, test cases, risks, regulations, defects, etc.).
- supporting multiple relationship types (not just “*satisfies*” or “*tests*”, but also “*conflicts with*”, “*depends on*”, “*refines*”, “*duplicates*”, etc.).
- providing contextual meaning, enabling you to understand why the link exists, what kind of relationship it is, and how strong or weak the relationship is.
- allowing for bidirectional navigation and impact analysis — so if one artifact changes, one can see all the related artifacts and assess ripple effects.
- use of visualizations (graphs, matrices, networks) instead of flat lists, helping teams spot gaps, overlaps, and dependencies.

These are very important principles, which are fundamental to any well-structured safety argument. GSN does provide for this (Dick 2002) to some extent but as it is only a notation, it needs a set of rules to supplement it — hence, for example, the good-practice principles for structuring safety arguments and evidence, set out at Appendix A hereto.

4.3 Application to Safety Requirements

Figure 5 below is from Fowler and Pierce (2012) and shows how the Jackson (1995) model of Figure 4 above can easily be applied to a safety environment, as follows:

- the *Application Domain* now covers the IEC 61508 concept of an *EUC*, and the *Operational Environment* in which it is deployed — each has its own *Domain Knowledge* ($\mathcal{D}$), which we know (or assume) to be true;
- the term *System* (comprising equipment, people and procedures) replaces the *Machine*;
- the primary Satisfaction Argument ($\mathcal{D}, \mathcal{S} \vdash \mathcal{R}$) remains the same, but a Design argument ($\mathcal{A} \vdash \mathcal{S}$) has been added¹⁶; and
- although not shown explicitly, the *Requirements* would include the basic safety targets that would need to be satisfied in the *Operational Environment*, e.g. tolerable maximum risk of accidents and/or hazards in the *Operational Environment*.

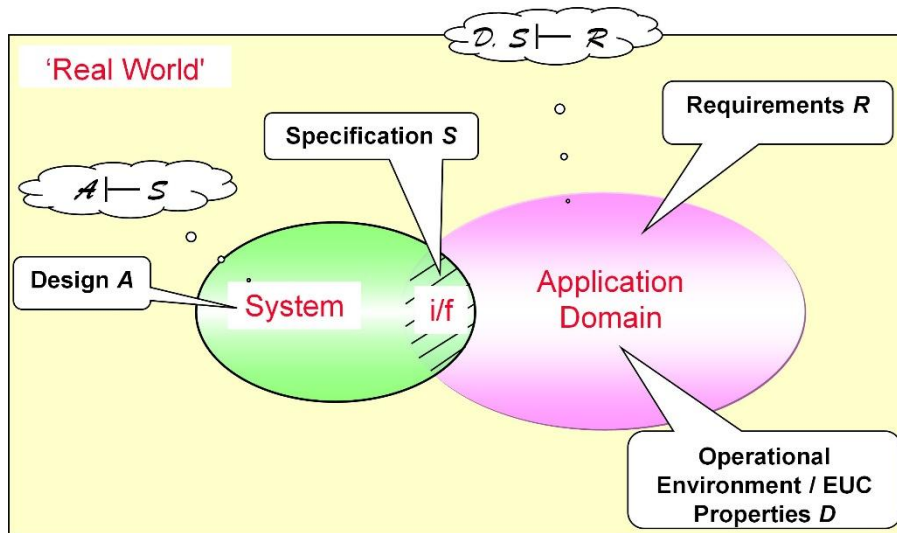


Figure 5 ~ Safety Interpretation of the Jackson Requirements-capture Model

It is this model that is used in developing Argument 1 and Argument 2, in Sections 5 and 6, respectively, below.

Before that, however, it would be worth pausing to consider three egregious past examples of where the ignorance of the above principles and practices appears to have led to disastrous consequences.

4.4 Some Lessons from History

The examples chosen to illustrate in this Sub-section the perils of ignoring the above principles and practices are:

- the Ariane 5 rocket explosion;
- the Überlingen mid-air collision; and
- the failure of a major London Underground Ltd. (LUL) re-signalling project.

¹⁶ $\mathcal{A}$ is Design; $\mathcal{D}$ has already been used for Domain Knowledge.

It is important to note, however, that *none* of these case studies are intended to be complete discussion of everything that went wrong — rather, they are focussed on those causal factors that can be seen to have strong links with the themes in this article.

4.4.1 Ariane 5 Rocket Explosion

This first example involves the maiden flight of the Ariane 5 rocket (Ariane 501), on 4th June 1996, which failed catastrophically within 40 seconds after initiation of the flight sequence, and which incurred a direct cost of at least \$370 million.

According to Dowson (1997) the sequence of events can be summarised as follows:

- *“An outside-expected-range value, derived from the launcher's actual horizontal velocity, led to an ‘unhandled exception’ in the Inertial Reference System (SRI)¹⁷ software. This caused the backup “hot standby” SRI to fail approximately 37 seconds after launch initiation (H0).*
- *Seventy-two milliseconds later, the active SRI (running identical software) failed for the same reason. [a common-mode failure]*
- *In the absence of a working backup, the failure of the active SRI caused it to transmit diagnostic bit patterns to the main On-Board Computer.*
- *The On-Board Computer, misinterpreting the diagnostic input as valid data, ordered full nozzle deflections of the solid boosters and the Vulcain main engine.*
- *These deflections created aerodynamic loads that separated the boosters from the main stage, (correctly) triggering the launcher's self-destruct system at approximately H0+40 seconds”.*

The Inquiry Board Report (Lions 1996) concluded, in essence, that poor software engineering practices were responsible for the *“software failure”* that caused the catastrophe; however, both Mark Dowson (1997) and Nancy Leveson (2001) disagree with the IRB’s use of *“software failure”*, since the software did exactly what it had been designed to do. Dowson (1997), in particular, offers a *“closer analysis of the Inquiry Report”* that *“reveals a rather different picture”*, as follows:

1. *“... the early part of the flight trajectory of Ariane 5 differs from that of Ariane 4 and results in considerably higher horizontal velocity values.”* (Inquiry Board Report, page 4).
2. *“The design of the Ariane 5 SRI is practically the same as that of an SRI which is presently used on Ariane 4, particularly as regards software.”* (Inquiry Board Report, page 3). *“...As should now be obvious, the software functioned precisely according to its specification and design, and no additional software-related measures — such as ensuring that all code explicitly included a handler for “others” conditions, improved QA, e.g. better test coverage, or use of improved approaches to software fault tolerance — could have ensured that some similar catastrophe did not occur.”*

Dowson then goes on to say that there are *“then, ... two astonishing aspects to the Inquiry Report”*; viz.:

¹⁷ French initialism

1. “... the report reveals that the expected flight trajectory of Ariane 5 was **deliberately** excluded from consideration in the design of a key component involved in its control; and
2. “... the Inquiry Board (which included software engineers) characterized the cause of the consequent flight failure as a software problem.

“Unfortunately, it is all too easy to imagine how the first of these happened in an industrial context: pressures of time and budget, if-its-not-broke-don't-fix-it arguments, etc. The second is harder to comprehend, and perhaps it is wiser not to speculate on its causes without better insight into the procedures followed by the Inquiry Board.

*So, how should we characterize the causes of failure of Ariane flight 501, and how can future such failures be avoided? At the very least, the failure was a systems engineering failure. As we all know, software can't be designed in isolation, and the system that it is part of includes its operating environment. ...A well-defined (and well designed) process for engineering systems such as Ariane 5 should make it impossible to exclude any significant aspect of the system **operating environment** from consideration. ...A good process needs to regulate not only how systems are designed and developed, but also how high-level decisions about that design and development are arrived at. Only by adopting and following a process that encompasses these considerations, will sound engineering practice be able to triumph over immediate pressures of time and budget”.*

In other words, failure to understand that, even if the *Requirements, Specification and Design* of the software for Ariane 5 did not change from those of Ariane 4, **changing the properties of the Application Domain**, i.e. of the rocket, could (and DID!) negate the satisfaction arguments $\mathcal{D}, \mathcal{S} \vdash \mathcal{R}$, with catastrophic results!!

4.4.2 Überlingen Mid-air Collision

According to the subsequent accident-investigation report (BFU 2004), a Boeing 757 (B757) cargo aircraft flying north, in cruise, at flight level (FL) 350, requested FL 360, and was incorrectly cleared to climb to this level. This created a crossing conflict with a Tupolev Tu-154 passenger aircraft¹⁸ flying on an approximately westerly track at the same flight level. For various reasons, detailed by BFU (2004), the Air Traffic Controller in the Zürich centre did not notice this conflict for some time. When the Controller did eventually notice the conflict, he instructed the Tu-154 aircraft to descend to FL 350 thus: “BTC 2937...descend flight level...350, expedite, I have crossing traffic”. Seven seconds after the start of this instruction, the ACAS on each aircraft issued a Resolution Advisory (RA). The B757 received a descend RA, and the Tu-154 received a climb RA, i.e. in the opposite direction to the Controller’s instruction. The B757 flight crew promptly obeyed their RA while the Tu-154 flight crew, after a short on-board discussion, chose to obey the Controller’s instruction.

The B757 flight crew reported their RA (badly!) 23 seconds after the RA was issued, and 16 seconds after the Controller had issued a second (more urgent) instruction to the TU154 to “expedite descent”. The Tu-154 flight crew did not report the RA, but did acknowledge the Controller’s second descent instruction, in response to which the Controller confirmed that “... we have traffic at your 2 o’clock position, now at [FL]360”. Both aircraft continued

¹⁸ Forty-six of the passengers on the TU154 were Russian schoolchildren on a school trip organised by a local UNESCO committee to the Costa Daurada beach area of Catalonia. in Spain.

to descend and collided. Around 20 seconds before the collision, the B757 ACAS issued an increase descent RA and at 8 seconds the Tu-154 ACAS issued an increase climb RA. At no stage during the sequence did ACAS generate a reversal command to either aircraft.

BFU (2004) correctly focussed more on the many egregious failings on the part of the responsible Swiss authority (skyguide¹⁹), the management at the Zurich Control Centre, and the (sole) Controller on duty at the time, than on why the ACAS, as a “socio-technical system”, failed to prevent the accidents despite the fact that the technical functioning of the ACAS *equipment* in both aircraft had been correct throughout. It is the latter question that occupied the minds of EUROCONTROL experts after the accident; leading to publication of many reports, including the final version of an argument-based ACAS Safety Case (EUROCONTROL 2011), which concerns us here.

During the development of the Safety Case, it became clear that ACAS operations were defined in a considerable number of disparate documents promulgated by ICAO, other international / national regulatory and standards bodies, ATM service providers, aircraft operators, and manufacturers (Thomas and Fowler 2013).

The problem was that, even at the highest level of definition, each of these documents captures only an individual aspect of ACAS operations. Consequently, there existed no conceptual description of ACAS operations as a whole that could be used as a starting point for assuring that the detailed provisions were themselves correct, coherent and consistent with each other, or for verifying that any future changes to ACAS operations would be compatible with the original intent of the system — in the terms of this article, what was needed, but was *totally absent*, was a *Specification for ACAS*.

The solution was to *reverse-engineer* the ACAS document set to yield a coherent definition of ACAS operations, including a set of basic operational rules, some examples of which are shown in Table 1 below (Thomas and Fowler 2013).

Table 1 ~ Examples of ACAS Basic Operational Rules

Ref.	Safety Objective
1	In the event of an imminent mid-air collision, one or both of the aircraft involved shall initiate a vertical manoeuvre such that the collision is avoided
2	Collision-avoidance manoeuvres shall be initiated in a timely manner and in a way that does not generate positive or negative ‘g’ forces that would over-stress the airframe or injure the aircraft occupants
3	Collision-avoidance action shall be unaffected by interference from air traffic services
4	Collision-avoidance action shall not interfere with the provision of air traffic services to non-involved aircraft
5	Collision-avoidance action shall take priority over the need to resolve situations that present a lower risk to the aircraft

An important part of the safety case work was, therefore, to check compliance of the existing ACAS design — especially the operational procedures — against the rules. Unsurprisingly, given that those procedures were written without the benefit of the rules, i.e. a Specification,

¹⁹ Swiss Air Navigation Services Ltd.

the analysis revealed both a number of serious non-compliances with the rules; and inconsistencies within the procedures themselves — two examples of which are as follows:

Firstly, the fact that, although Controllers were told not to intervene in the event of an RA (ICAO 2007), the only way they could know that an RA was happening was by a report from the Flight Crew, and yet such a report was low down in the Flight Crew's priorities during an RA event (ICAO 2006).

Secondly, Flight Crew procedures (ICAO 2006) state, in Section 3.1.2, that:

“Nothing in the procedures specified in section 3.2 ...shall prevent pilots-in-command from exercising their best judgement and full authority in the choice of the best course of action to resolve a traffic conflict or avert a potential collision”,

whilst section 3.2 itself clearly states that:

“... in the event of an RA, pilots shall:

“1) respond immediately by following the RA as indicated, unless doing so would jeopardize the safety of the aeroplane;

“2) follow the RA even if there is a conflict [sic] between the RA and an air traffic control ... instruction to manoeuvre”.

Sadly, Überlingen was not an isolated incident. Just 18 months before the Überlingen collision, a conflict occurred between two Japan Airlines (JAL) aircraft, a Boeing 747 climbing outbound from Tokyo to Okinawa, and a DC-10 inbound to Tokyo from South Korea near the city of Yaizu. In response to an STCA²⁰ alert, the air traffic controller instructed the B747 to descend, about three seconds before ACAS RAs were issued in both aircraft. The ACAS equipment on board both aircraft functioned correctly, giving a climb RA to the descending B747, and a descend RA to the DC-10. The DC-10 flight crew obeyed the descend RA, but the B747 crew continued its descent, in accordance with the prior air traffic controller instruction, but in contradiction to the climb RA (EUROCONTROL 2022). Fortunately, a mid-air collision was avoided, but only by a few feet!

EUROCONTROL (2011) poses the rhetorical question: *“Would it be unreasonable to suggest that the actions of the B747 pilot in the Yaizu incident and of the TUI54 pilot in the case of the Überlingen accident ... as falling within what is permitted by PANS-OPS section 3.1.2 rather than contravening PANS-OPS section 3.2?”*

From the way that the above rules are expressed (especially their complete independence from the technical solution), it should be apparent that, in terms of this article:

- the Rules would form a major part of the *Specification*, if one had existed;
- the Aircraft in the airspace (*Operational Environment*) would be the EUC; and
- Air Traffic Services would be part of the *EUC Control System*.

The problem was that ACAS had been brought into service, and had spent many years *in* service, without the benefit of a *Specification*, seemingly allowing operating procedures, i.e. a vital part of the wider-system *Design*, to be developed without an appropriate degree of rigour.

It has to be said, though, that whilst the post-Überlingen safety analysis produced a set of rules, which enabled inconsistencies in the procedures to be revealed, we will see, in Sub-

²⁰ Short-Term Conflict Alert (STCA) is a radar-based tool that warns air traffic controllers of aircraft being too close to each other.

section 5.4 below, that it is crucial that analysis techniques be applied in order to ensure that the *Specification* itself is produced with the appropriate degree of rigour, in terms of its correctness, completeness, and coherency. Unfortunately, it seems that ACAS was introduced by the Commercial Aviation sector without the benefit of a proper safety assessment, on the basis that — as a safety net, of last resort — it would be better to have ACAS than not have it at all. Such a naïve view would of course overlook the possibility of ACAS creating a new *system-generated* hazard, due to dysfunctional operation of the wider system — a hazard that would not otherwise have occurred.

A clear case of failing to appropriately balance “*the competing demands and attractions*” of the world and the machine (Jackson 1995), with tragic consequences.

4.4.3 Railway Signalling Contract Abandonment

4.4.3.1 Background

The following is a direct quote from an article in the International Railway Journal, of 3rd January 2014 (IRJ 2014):

“London Underground (LUL) and Bombardier Transportation have announced that a contract for re-signalling of the [Sub-Surface Railway (SSR)] comprising the Circle, District, Hammersmith & City, and Metropolitan lines with communications-based train control (CBTC) will be relet”

“Bombardier [had been] awarded the £354m contract in June 2011 and was expected to supply its Cityflo 650 moving block automatic train control system which it has already successfully deployed on lines 1 and 6 of the Madrid Metro, increasing capacity by 30%...”

“However, Transport for London (TfL) says [in Jan 2014] as the work progressed it became clear that the fit between the system offered and the Underground network is such that the contract is not going to deliver what is needed, in the time required, while remaining cost-effective. “We have been working closely with Bombardier to find a way forward on what is one of the most challenging and complex pieces of work on the Tube,” says Mr Mike Brown, LUL’s managing director. “However, it has become apparent to both parties that for the work to be completed within or close to the planned deadline, we need to push on with works with another contractor....”

In August 2015, LUL announced the award of a £760m contract to the Thales Group to “modernise the signalling and train control system on [LUL’s] Circle, District, Metropolitan and Hammersmith & City lines” (IRJ 2015).

In March 2016 the London Assembly, Budget and Performance Committee, issued a scathing report (London Assembly 2016) on the project failure, which, in the words of its chairman (John Biggs) concluded that, inter alia:

“TfL has grossly mismanaged its signalling contract with Bombardier, which means that anyone taking the District, Circle, Metropolitan, or Hammersmith & City lines will have to wait five years longer than originally expected to see the improvements. And there is a staggering cost associated with this mismanagement, which leaves TfL with £886 million less to spend on its capital programme than it thought it had”.

As a systems-safety engineer of nearly 20 years' experience, 18 months of which was as a member of the LUL SSR project safety-engineering team, prior to the cancellation of Bombardier's contract, the author of this article makes no comment on the above political and commercial aspects of the project but does offer the following personal insight into the technical problems that beset the project, from the perspective of this article²¹.

4.4.3.2 A Safety-engineering Viewpoint

It is self-evident that the primary purpose of any railway signalling system is to prevent accidents, and that the primary properties of such a system that determine its ability to prevent accidents are those related to its *functionality and performance*. It should therefore be equally self-evident that producing assurance of correct functionality and performance of a safety-related system is a vital part of any system safety assessment; it is also entirely consistent with the concept of safety functions in IEC 61508-1 and was a specific requirement of rail safety standard EN 50129, at the time, but *only* of Annex B, section B.2 thereto.

Of course, in specifying and designing a signalling system, it is necessary also to ensure that the (extremely low) likelihood of a signalling system causing, or failing to prevent, accidents is acceptable. The primary properties of a signalling system that determine this likelihood are its *reliability and integrity*; producing assurance of this is also a requirement of the above IEC and CENELEC standards.

Bombardier's 2011 SSR re-signalling contract with LUL was founded on supply of its CITYFLO650 system, with a safety-assurance strategy that seemed (from Bombardier's documents at the time) to be based on the following, unsubstantiated assumptions:

- that a system designed for use, and deployed, on Shenzhen and Madrid metros would meet the uniquely complex needs of the SSR with "*some upgrades and enhancements*";
- that evidence based on the development and in-service experience of Shenzhen and Madrid products would be of sufficient scope and quality to support *cross-acceptance* arguments for application to the SSR;
- that compliance with (generic, *process-based*) CENELEC standards would provide sufficient safety assurance for CITYFLO650-enabled, moving-block operations across the SSR network;
- that evidence of the functional correctness of a SIL 4 system design can be based almost entirely on a simple DOORS-based traceability process, and testing, without rigorous design analysis.

By 2013, it was becoming increasingly clear, to LUL's SSR safety team at least, that none of the above assumptions were valid, and that, in relation to the thrust of this article, Bombardier's approach to the project was seriously flawed, as follows:

Firstly, Bombardier had little or no understanding of the unique complexity of the SSR *Application Domain*, i.e. the EUC, its *Control System* (the existing railway infrastructure) and their *Operational Environment*, compared with the very basic Madrid and Shenzhen metros, until LUL (somewhat belatedly), produced a detailed description thereof.

²¹ This insight is a summary of a highly critical technical report, written by the author of this article, and submitted to LUL two months before the Bombardier contract was cancelled.

Secondly, in accordance with its contract, Bombardier developed a very detailed structured Safety Argument, for the introduction of CITYFLO650 to the SSR. That was a somewhat strange responsibility to place on an *equipment supplier* since the responsibility for the assuring the safety of SSR *operations* would have lain with LUL as the service operator. Worst still, the Safety Argument was based almost entirely on compliance with the processes of the extant CENELEC rail safety standards, with the result that:

- it contained no valid claims for the functional safety of the CITYFLO650 *products*;
- it was based on safety-assessment processes that addressed only the reliability of the CITYFLO650 products, to the exclusion of their safety functionality and performance;
- it failed to recognise the crucial role that the properties of the *Application Domain* play (or should play) in a system safety argument (see Arg 1.2 at Sub-section 5.3 below).

Thirdly, apart from some high-level operational requirements provided by LUL, there was no equivalent of a *Specification* (see Arg 1.3 at Sub-section 5.4 below) or high-level Design Analysis (see Arg 2 at Section 6 below) — in fact, no functional-safety requirements existed, at any level, for the application of the CITYFLO650 system to the SSR. Bombardier did eventually produce a detailed set of textural functional-requirements statements that might have been applicable at the physical-Design level, but it was entirely devoid of any analysis and any requirements covering system behaviour.

Fourthly, much of Bombardier’s product-level safety-assurance strategy seems to be based on the *Cross-acceptance* of individual products from their Shenzhen and/or Madrid baselines. Whilst such an approach was allowed in principle under CENELEC standards, the requirements for cross-acceptance, as laid down in guidance document CLC/TR 50506, sections 4.4.1a) to d), f), and g), were so demanding that it is unlikely that any of Bombardier’s products (with the possible exception of *Ebilock* interlocking) would have been suitable for Cross Acceptance. Indeed, an independent “gap analysis” of CITYFLO650 software in 2012 regarding the SIL 4 RATP and VATP sub-systems’ software from the Shenzhen baseline, revealed critical shortcomings in the software-development processes.

In 2012-13, in a serious attempt to correct these *ab initio* flaws, LUL’s SSR project safety team set out to:

- produce its own, rigorous Safety Argument for the introduction of CITYFLO650-enabled, moving-block operations on the SSR. In outline, this argument was similar to that presented herein, but was further decomposed in full detail, across the whole of the safety lifecycle;
- produce a detailed description of the *Application Domain* for the SSR, which showed *inter alia* that, because of track sharing between the SSR, other LUL lines and some Network Rail lines, the CITYFLO650 system would have to: be rolled out across 12 so-called Migration Areas; accommodate 8 different train types; support 5 different ATP systems; and interface with 6 different signalling systems, across at least 15 signalling boundaries; and
- carry out an operational hazard analysis and develop, what are known herein as, a full Specification and a technology-independent, high-level Design, addressing more than 100 operational scenarios and covering *normal*, *abnormal* and *failure conditions*. The article by Fowler and Graebner (2023) is based almost entirely on that work.

Regrettably, all this came too late to save the project and served only to highlight the yawning gap between the system offered by Bombardier and what the SSR required in order

to meet its operational and safety needs, leading to cancellation of the contract on 31st December 2013.

It could be said that, from the outset, LUL's and Bombardier's approach to the project seems to have been focussed mainly on how the existing CITYFLO650 system could be used on the SSR, and nowhere near enough on first understanding what that system would have to do in order to meet the operational and safety needs of the SSR — this imbalance was completely at odds with the warning of Jackson (1995) that “*Failure to balance [the competing demands and attractions of the World and the Machine] harms our work*” — quite an understatement, in the context of LUL's loss of £886 million for zero benefit to its business and its customers (London Assembly 2016)!

In Sections 5 *et sequitur*, Arguments 1 and 2 are developed in more detail, based on the principles and lessons discussed above, and the results used to assess the safety-assurance role and capabilities of IEC 61508-1 in the specification of safety-related systems for the transport sector.

5 Decomposition of Arg 1

5.1 Preamble

Unsurprisingly, the first level of decomposition of Arg 1 comes straight from the *satisfaction argument* of Hammond *et alia* (2001), set out in Sub-section 4.1 above, and the logical relationship $\mathcal{D} . \mathcal{S} \vdash \mathcal{R}$ first proposed by Jackson (1995).

For that relationship to hold, and thus for Arg 1 to be true, we must show that each of the three sub-arguments, as captured in *adapted* GSN form in Figure 6, is true. These sub-arguments are discussed in detail in Sub-sections 5.2 to 5.4, respectively.

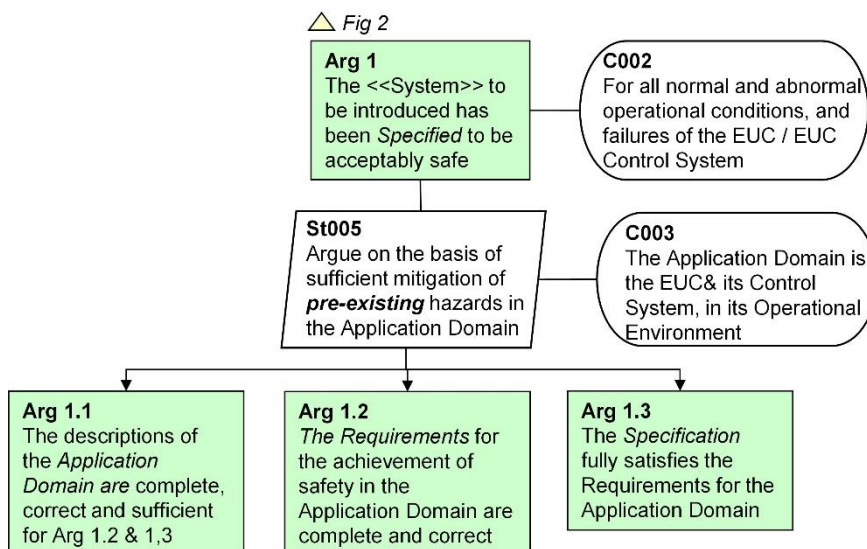


Figure 6 ~ First Level of Decomposition of Argument 1

It is important to note that the $\mathcal{D} . \mathcal{S} \vdash \mathcal{R}$ relationship is three-way, and that the process of satisfying Arg 1, through satisfaction of sub-arguments Args 1.1 to 1.3, is necessarily iterative — see also the discussion on rich traceability in Sub-section 4.2 above.

5.2 Arg 1.1 Domain Knowledge — Completeness & Correctness

5.2.1 Strategy

The *Strategy* for satisfying Arg 1.1 is to provide arguments and evidence which show that the properties of the *Application Domain* have been identified and documented completely, correctly and sufficiently, to support the satisfaction of Arg 1, at this stage — and later to support the satisfaction of Arg 2 to Arg 5.

In IEC 61508-1 terms, it would be necessary to show that inherent properties of the following Application Domain entities have been described fully and correctly, and are sufficient to support the *Requirements* in Arg 1.2 and the *Specification* in Arg 1.3:

- the EUC, e.g. the movement of aircraft through the airspace (Fowler and Fota 2023) or the movement of trains around the railway (Fowler and Graebner 2023);
- the Operational Environment and the EUC Control System; and
- all of the national and international safety regulations that are applicable to the subject of the overall Safety Argument.

5.2.2 Assurance Objectives for Arg 1.1

Credible arguments and evidence shall be provided to demonstrate the achievement of each of the ten Assurance Objectives defined in the pair of tables below.

Table 2 ~ Direct Assurance Objectives for Arg 1.1

ID	What has to be Demonstrated
1.1.1	The Domain properties address all the needs of the system Specification
1.1.2	The properties, and their attributes, accurately and completely reflect the actual (or assumed) properties of the Application Domain
1.1.3	Each Application Domain property description includes either: • a specification for each of the appropriate attributes of the property, or • a valid justification that the attribute is not applicable
1.1.4	The properties, and their attributes, form a coherent set and are specified in sufficient detail and clarity to enable the system Specification, Design and Implementation to achieve the required level of safety

Table 3 ~ Backing Assurance Objectives for Arg 1.1

ID	What has to be Demonstrated
1.1.5	The form of representation of the properties is appropriate to the Application Domain and allows an adequate determination of the Domain Requirements and/or system Specification and subsequently of the system high-level Design and Implementation, as applicable

ID	What has to be Demonstrated
1.1.6	The methods and techniques used are appropriate for the type, and safety-integrity level, of the properties / attributes concerned
1.1.7	The methods and techniques used are capable of supporting the identification of all properties of the Application Domain under all conditions: i.e. <i>normal</i> , <i>abnormal</i> and <i>failure</i> states of the EUC / EUC Control System
1.1.8	The methods and techniques have been applied by suitably qualified and experienced staff <i>Note: staff are deemed to be suitably qualified and experienced if they understand the methods used, are experienced in using them, and understand the required properties / attributes, and the operational and system contexts</i>
1.1.9	Any tools, used in the identification, description and configuration management of the Application Domain properties, have been verified and validated, as appropriate to the safety-integrity level of the properties concerned
1.1.10	A process that is independent of the means by which the properties were derived in the first place has demonstrated the validity of the properties, as appropriate to the safety-integrity level of the properties concerned.

5.2.3 Supporting Evidence for Arg 1.1 from IEC 61508

IEC 61508-1 states, in its subclause 7.2.1, that the “*objective [sic] of the requirements of this subclause is to develop a level of understanding of the EUC and its environment (physical, legislative, etc.) sufficient to enable the other safety lifecycle activities to be satisfactorily carried out*”.

It then sets out three “requirements” [sic] that are relevant here ²²:

“7.2.2.1 A thorough familiarity shall be acquired of the EUC, its required control functions and its physical environment.

7.2.2.4 Information about the current safety regulations (national and international) shall be obtained.

7.2.2.6 The information and results acquired ... shall be documented”.

In assurance terms, this is quite weak. Whereas some indication is given about what needs to be done, i.e. the Assurance *Activities* involved in deriving the Application Domain properties, there is nothing in the IEC 61508-1 text (subclause 7.2.1) that requires the completeness and correctness of those properties to be demonstrated, as required by the much more stringent Assurance *Objectives* of Sub-section 5.2.2 above.

5.2.4 An ATM Perspective on Arg 1.1

Sub-section 3.1 of Fowler and Fota (2023) provides a summary of the range of data that would be required initially in order to characterise the Operational Environment, the EUC and the EUC Control System for the Point Merge operational concept. Because that ATM

²² Three other requirements are set out in IEC 61508-1, subclause 7.2.2; however, as they refer to the identification of hazards, they are covered herein under Arg 1.2.

application is a generic view of Approach operations, rather than a case study for a specific airport, the range of data is necessarily illustrative; however, the reference (*unlike* IEC 61508-1) does emphasise the need for the Application Domain data, and their properties, to be verified for completeness, correctness, consistency and sufficiency to support Arg 1.3 and, subsequently, Arguments 2, 3, 4 and 5.

With those provisos, Sub-section 3.1 of Fowler and Fota (2023), offers a reasonable illustration of what would need to be done in order to provide adequate Evidence in support of Arg 1.1 for a typical new ATM operational concept. In reality of course:

- the actual data would need to be more detailed and specific to the airport concerned. However, for the purposes of satisfying Arg 1, only data related to the Specification would be required at this stage in the safety-assessment process — data related the Design and the Implementation of the System would be required later, in support of Arg 2 and Arg 3 respectively; and
- the completeness, correctness, and consistency of the actual data would need to be demonstrated, to the level of integrity — in IEC 61508 terms, the Safety Integrity Level (SIL) — required of the data, in the System context.

Furthermore, all data would need to be maintained, throughout all phases of the safety lifecycle outlined in Figure 3 above.

5.2.5 A Rail Perspective on Arg 1.1

Sub-section 4.1 of Fowler and Graebner (2023) provides a summary of the range of data that would be required initially in order to characterise the Operational Environment, the EUC and the EUC Control System. Because the Railway-signalling application is a generic view of Moving Block operations, rather than a case study for a specific railway, the range of data is necessarily illustrative; however, the reference (*unlike* IEC 61508-1) does acknowledge the crucial need for the Application Domain data, and their properties, to be verified for completeness, correctness, consistency, and sufficiency in support of Arg 1.3 and, subsequently, Arguments 2, 3, 4 and 5.

It should be noted, though, that there is an error in Sub-section 4.1 of Fowler and Graebner (2023), in that track and station parameters should have been listed under the “EUC Control System” in Sub-section 4.1.4, rather than as properties of the (Operational) Environment in Sub-section 4.1.3.

With those provisos, Sub-section 4.1 of Fowler and Graebner (2023), provides a reasonable illustration of what would need to be done in order to provide adequate Evidence in support of Arg 1.1 for a typical new moving-block railway-signalling system.

In practice the full data set would be huge, and would probably include items that would require a SIL 4 attribution; and assurance of the completeness, correctness, coherence, and integrity of the whole data set, throughout the development and operational life of the CBTC System, would be a significant challenge.

5.3 Arg 1.2 Application Domain Requirements

5.3.1 Strategy

Traditionally, *Requirements* for the safety of operations are mandated by the relevant regulatory body, rather than by the responsible operating authority, and such *Safety Targets* would usually relate only to a maximum tolerable likelihood of Harm²³.

The usefulness of such targets, in *a priori* safety assessments is quite limited because of the *a posteriori* nature, and necessarily extremely low frequency of occurrence, of such harmful events. A far more practicable approach would be to place limits on the frequency of occurrence of the Hazards such that the regulatory Safety Targets for Accidents would thence be met. Furthermore, as IEC 61508-1²⁴ makes very clear, Hazards are associated with the EUC, and it follows that the tolerable rates of occurrence of the Hazards must fall within the Application Domain Requirements, not within the System Specification (Arg 1.3).

***Note:** Further justification for the inclusion of such pre-existing Hazards here can be seen from an example of a Train Derailment accident, for which Fowler and Graebner (2023) states that the hazardous precursor would be “HP#3 - Train speed exceeding capabilities of the track infrastructure...”; i.e. the occurrence of the hazardous state, HP#3, depends on **two** properties: the **speed** at which the train, i.e. the EUC, is travelling and the **physical properties** of the track (i.e. part of the EUC Control System), e.g. its curvature, that define its ability to accommodate a train travelling at that speed. It is asserted, therefore, that what IEC 61508-1 describes as EUC Hazards are actually emergent properties of the Application Domain, not properties of the System; the only exception to this are hazards caused by incorrect or spurious operation of the System — as happened in the case of the Überlingen mid-air collision, outlined in Sub-section 4.4.2 above.*

Therefore, the *Strategy* for satisfying Arg 1.2 is to provide arguments and evidence which show that:

1. a valid set of *Safety Targets*, which specify the maximum tolerable risk of an accident, have been determined, in accordance with national regulatory requirements;

Note: The Safety Targets could be specified for all accidents together or for each accident type separately.

2. all potential hazards associated with the EUC, in its Operational Environment, covering all accident types, and within the scope of all potential risk-reduction measures, have been identified;
3. the probability that each Hazard, if not mitigated, would lead to an accident has been determined;
4. the tolerable hazard occurrence rate (THOR) for each and every identified Hazard, has been determined such that the corresponding Safety Target is met.

²³ Defined in IEC 61508-4 as “...death, physical injury or damage to the health of people or damage to property or the environment”.

²⁴ For example, in IEC 61508-1 subclause 7.4.3.2.

Hence the *Requirements* for the safety of operations in the Application Domain comprise simply a description of, and THOR for, each and every in-scope EUC Hazard.

5.3.2 Assurance Objectives for Arg 1.2

Credible arguments and evidence shall be provided to demonstrate the achievement of each of the 15 Assurance Objectives defined in the tables below.

Table 4 ~ Direct Assurance Objectives for Arg 1.2

ID	What has to be Demonstrated
1.2.1	Safety Targets, which state the maximum tolerable risk of an accident, have been specified
1.2.2	The Safety Targets have been determined, in accordance with the relevant national regulatory requirements
1.2.3	The Safety Targets have been derived for all accidents as a whole, or for each accident type separately, as appropriate
1.2.4	All potential hazards associated with the EUC, in its Operational Environment, covering all accident types, and within the scope of all potential risk-reduction measures, have been identified
1.2.5	The consequences associated with each of the identified Hazards have been determined in terms of the potential accident outcome(s)
1.2.6	The probability of each unmitigated Hazard leading to an accident has been determined
1.2.7	The hazard-analysis process has addressed all reasonably foreseeable circumstances
1.2.8	The hazard-analysis results are sufficient to support the derivation of the Application Domain safety <i>Requirements</i> and the System safety <i>Specification</i>
1.2.9	For each and every identified Hazard, a THOR has been derived such that the Safety Target for the related Accident(s) could be met

Table 5 ~ Backing Assurance Objectives for Arg 1.2

ID	What has to be Demonstrated
1.2.10	The methods and techniques used in the derivation and verification of the Safety Targets are appropriate to the Application Domain
1.2.11	The processes, tools and techniques that are used in the Hazard analysis, are appropriate to the Application Domain, and allow an adequate analysis of the THORs
1.2.12	The methods and techniques used in the derivation and verification of the THORs are appropriate to the Application Domain, and allow an adequate analysis of the <i>Specification</i>
1.2.13	All methods, processes, tools and techniques used in the development of Direct evidence for Arg 1.2 are appropriate to the specified safety integrity levels

ID	What has to be Demonstrated
1.2.14	All methods, processes, tools and techniques used in the development of Direct evidence for Arg 1.2 have been correctly applied
1.2.15	Only suitably qualified and experienced staff have applied the methods, processes, tools and techniques used in the development of Direct evidence for Arg 1.2

5.3.3 Supporting Evidence for Arg 1.2 from IEC 61508

IEC 61508-1, clause 7.4, clearly recognises the need for a Hazard and Risk Analysis to be carried out, and sets three “*objectives*” [sic] for it:

1. “... to determine the hazards, hazardous events and hazardous situations relating to the EUC and the EUC Control System, in all modes of operation, and for all reasonably foreseeable circumstances ...”;
2. “... to determine the event sequences leading to the hazardous events ...”; and
3. “... to determine the EUC risks associated with the hazardous events ...”.

Being a generic, multi-sector safety standard, IEC 61508-1 cannot be expected to detail how a tolerable level of risk would be determined, despite the need for knowing this being crucial to its fundamental concept of Necessary Risk Reduction”. Thus, while the first two “*objectives*” [sic] strongly imply the need to demonstrate the completeness and correctness of the Hazard and Risk Analysis results, the overall process involved lacks the context of what would be a tolerable level of risk for the EUC and a tolerable rate of occurrence of each of EUC hazard — thus there is little in the IEC 61508-1 text (clause 7.4) that explicitly requires the completeness and correctness of the Hazard and Risk Analysis results to be demonstrated, as required by the much more stringent Assurance Objectives of Sub-section 5.3.2 above.

The third objective seems to be related to subclause 7.5.2.4 of IEC 61508-1, which states that:

- “...overall safety integrity requirements shall be specified in terms of either:
- the risk reduction required to achieve the tolerable risk, **or**
 - the tolerable hazardous event rate so as to meet the tolerable risk”.

Since the strategy for Arg 1.2, set out in Sub-section 5.3.1 above, is based on the latter of these options, objective #3 above is not required; indeed, subclause 7.4.2.10 of IEC 61508-1 helpfully recognises the need to “... consider... the tolerable risk for each hazardous event”.

5.3.4 An ATM Perspective on Arg 1.2

Sub-section 3.3 of Fowler and Fota (2023) describes five accident types relevant to Approach airspace, which has been adapted from ICAO (2011)²⁵ and, in each case, involves

²⁵ ICAO (2011) is intended for use in a *posteriori* categorisation of actual occurrences, rather than a *priori* safety assessment—hence the need for some adaptation

death or serious injury to one or more of those on board. It then presents (from SESAR (2018b)) five related EUC Hazards²⁶ and a mapping back to the associated accident types.

Because Fowler and Fota (2023) present a *generic* view of Point Merge operations, rather than a specific case study, it does not quantify either the probabilistic relation between the Hazards and Accidents, or the Tolerable Hazard Occurrence Rate for each Hazard. Instead, it refers to the Accident Incident Models of SESAR (2018a), from which those parameters could be derived.

Nevertheless, Sub-section 3.3 of Fowler and Fota (2023), does provide a reasonable illustration of what would need to be done in order to provide adequate Evidence in support of Arg 1.2 for a typical new ATM operational concept, at a specific existing airport.

5.3.5 A Rail Perspective on Arg 1.2

Sub-section 4.3 of Fowler and Graebner (2023) describes the various accident types that are relevant to railway operations, and, in each case, would probably involve death or serious injury to one or more of those on board a train, or to the workforce or members of the public on, or in the vicinity of, the track. It presents a comprehensive list of the EUC-related Hazards that are likely to fall within the scope of the System-to-be-specified, in relation to what are seen to be the most credible accident outcome(s). The accident and hazard sets were both derived from doctoral research carried out by Ivan Lucic, and adopted for general use by London Underground Ltd. (Lucic 2015).

Unfortunately, in sticking (perhaps too closely!) to the related IEC 61508-1 processes at this stage, the reference has little evidence to offer in relation to deriving Tolerable Hazard Occurrence Rates; rather, it defers the matter (incorrectly) to the subsequent discussion, on Overall Safety Functions. Otherwise, Sub-section 4.3 of Fowler and Graebner (2023), provides a reasonable illustration of what would need to be done in order to provide adequate Evidence in support of Arg 1.2 for a typical new moving-block railway-signalling system.

5.4 Arg 1.3 Specification — Satisfaction of Requirements

5.4.1 Overview

Jackson (1995) explains that a *Specification* captures the properties that are *shared* by the Application Domain and the Machine, i.e. the System, and acts as a sort of logical interface between those two entities. In essence, it has to specify everything that the System has to *make happen* in the Application Domain in order to satisfy the Requirements thereof, with two provisos:

- it should contain *no* view or assumption regarding *how* the System might do that, in terms of its Design or Implementation, i.e. it should take a “black-box” view of the System; and
- it should be expressed in a way that the System *Design* could ensure that everything that the System has to make happen in the Application Domain, would actually happen.

A more challenging question that the author was faced with. in researching the subject of this article, was what form should a Specification take for transport applications, given that the safety Requirements for the Application Domain would most likely be expressed in the

²⁶ SESAR (2018b) uses the term “*pre-existing*” hazards for what IEC 61508-4 defines as EUC hazards.

form of THORs for each and every identified EUC Hazard. At first sight, it might appear that a Specification is synonymous with the concept of Overall Safety Functions (OSFs) of IEC 61508-1. However, in applying the OSF concept to the simple example of a pedestrian-safety problem, Fowler (2022) quite reasonably deduced that the main hazard would be “...moving traffic and pedestrians intending to occupy the same area of the road surface at the same time”, but ended up with a simple functional safety requirement that:

“In the area of potential conflict, the ... safety function shall ensure the safe separation of pedestrians and moving road traffic temporally and/or spatially”.

Whereas such a requirement would not preclude any particular mitigation strategy or ultimate technological solution and might be, at the highest level, sufficient in itself to mitigate the Hazard, it is not particularly enlightening since all it is really saying is don’t let the Hazard happen!

In order to avoid such anodyne outcomes, a somewhat different approach is proposed, as follows.

Firstly, any options that IEC 61508-1 defines as “*Other Risk-reduction Methods*” (ORRMs) should be separated out *before* more-focused and more-detailed requirements for mitigation of the Hazards are derived. This would be a minor departure from IEC 61508-1 but the reasoning for this can be seen in, say, the case of a choice between a pedestrian crossing and a pedestrian underpass²⁷, for which it would be largely pointless to try to derive a *single* set of high-level safety requirements to cover both options.

Secondly, in order to avoid straying into the area of system Design, the requirements in a Specification should be *rule-based*, rather than functional, i.e. stating what has to *happen* in the Application Domain, *rather* than stating *how* it has to be done. In ATM, these rules would include the *separation minima* for aircraft in the airspace concerned; in the rail sector, the rules would include the *signalling principles* for the railway concerned.

Thirdly, in order to be really meaningful, the *rules* would need to be set in the associated operational context, otherwise it would not be valid to argue that the Specification would be sufficient to satisfy the Requirements “*under all normal, abnormal and failure conditions*” (see Sub-section 5.4.2). In ATM, the question as to where and when separation *rules* should be applied would depend on the operational concept (e.g. Point Merge, as outlined in Fowler and Fota (2023)) and, within that, would vary between the various phases of flight. In the rail sector, there would be significant variation in the *rules* between, say, *fixed-block* and *moving-block* operations and, within that, on different parts of the railway. Specific examples of the application of the above will be provided in Sub-sections 5.4.5 and 5.4.6.

Note: although it is beyond the scope of this article to enter into detailed discussion about the introduction of AI into safety-related applications, it would be important, in such cases, to consider whether the AI would be used as a means of satisfying the existing Specification rules more efficiently or, say, ‘machine learning’ would be used as a means of developing and applying new, and more-efficient, rules. In the latter case, the safety assurance problem would clearly be very much greater!

²⁷ See Fowler (2022).

5.4.2 Strategy

The Strategy for satisfying Arg 1.3 is to show that the *Specification* correctly states what is necessary and sufficient to achieve tolerable safety, in the Application Domain. Specifically, it is necessary to provide arguments and evidence which show that:

1. The safety *Requirements* that are to be addressed by the *System* have been identified as such, and are *separate* from those requirements to be addressed by what IEC 61508-4 defines as ORRMs;
2. the *Specification* for the System states correctly what is necessary and sufficient to satisfy those Application Domain safety Requirements that have been allocated to the System, under all *normal*, *abnormal*, and *failure* states of the EUC, the EUC Control System and the Operational Environment, as well as the *transitions* between those states; and
3. the *Specification* states correctly what is necessary and sufficient, and in sufficient detail and clarity, to enable the subsequent system *Design* and *Implementation*, in conjunction with any identified ORRMs, to achieve the required level of safety in the Application Domain.

5.4.3 Assurance Objectives for Arg 1.3

Credible arguments and evidence shall be provided to demonstrate the achievement of each of the 10 Assurance Objectives defined in the tables below.

Table 6 ~ Direct Assurance Objectives for Arg 1.3

ID	What has to be Demonstrated
1.3.1	The Application Domain safety <i>Requirements</i> have been completely and correctly allocated and apportioned to the System and any ORRMs
1.3.2	Any assumptions about ORRM contributions to the mitigation of Hazards are clear, quantified and justified
1.3.3	At least one <i>Overall Safety Function</i> has been developed for each Hazard, clearly and accurately reflecting the System's required contribution to the mitigation of Hazards, at the Specification level <i>Note: the IEC 61508-1 concept of an Overall Safety Function is used here merely as a convenient way of expressing a high-level function generically – its use is not limited to the Standard's meaning of the term</i>
1.3.4	The rule-based Overall Safety Requirements derived for each Overall Safety Function are sufficient to mitigate the corresponding Hazards in the Application Domain Requirements
1.3.5	The Overall Safety Requirements set out in the Specification adequately specify the required safety behaviour of the System at the Specification level, for all normal, abnormal and failure states of the EUC, the EUC Control System and the Operational Environment, and the transitions between those states

ID	What has to be Demonstrated
1.3.6	A target Safety Integrity requirement has been specified for each Overall Safety Function, such that the corresponding THORs in the Application Domain safety Requirements would be met <i>Note: In this context, Safety Integrity is defined as the “...probability of a ... safety-related system satisfactorily performing the specified safety functions under all stated conditions, within a stated period of time” (IEC 61508-4)</i> <i>Thus, the form of these target Safety Integrity Requirements is entirely different from what IEC 61508-1 describes as “Safety Integrity Requirements”</i>

Table 7 ~ *Backing Assurance Objectives for Arg 1.3*

ID	What has to be Demonstrated
1.3.7	The analytic methods, techniques and tools used in the derivation and verification of the Overall Safety Requirements are appropriate for the attributes of those requirements
1.3.8	The analysis notations are appropriate to the Application Domain and the Safety Integrity requirement specified for each Overall Safety Function
1.3.9	The analysis notations allow an adequate analysis of the <i>Design</i>
1.3.10	Only suitably qualified and experienced staff have applied the analysis methods, techniques and tools

5.4.4 Supporting Evidence for Arg 1.3 from IEC 61508

As noted in Fowler (2020), the language of Section 7.5 of IEC 61508-1 can be quite difficult in places; nevertheless, the IEC 61508-1 concept of *Overall Safety Functions*, and (within them) *Overall Safety Requirements*, is consistent with the Jackson (1995) description of a *Specification*.

The importance of this IEC 61508-1 concept should not be underestimated; however, clause 7.5 of IEC 61508-1 does not offer nearly enough to satisfy the Assurance Objectives for Arg 1.3, listed in Sub-section 5.4.3 above, mainly because OSFs are defined at too high a level, i.e. at a level above which they are allocated to Safety Related Systems and/or Other Risk Reduction Measures. This means that the OSFs act as little more than a *basic-traceability* path between the *Application Domain Requirements* to mitigate Hazards (in Arg 1.2) and the SRSs that provide that mitigation (in Arg 2). The problem with that approach in relation to the transport sectors, is that the analysis needed to demonstrate *satisfaction* of the Requirements is made very much more complex and difficult if it is left entirely to the subsequent high-level *Design* phases of IEC 61508-1 clauses 7.6 and 7.7.

Without Jackson’s work, it would be all too easy to underestimate the enormous potential value that clause 7.5 of IEC 61508-1 *could* bring to the development of safety-related systems in the transport sector. Fortunately, there is much in clause 7.7 of IEC 61508-1 (SRS high-level *Design*) that could be adapted and applied to the earlier phase, i.e. to the development and validation of Overall Safety Requirements, in clause 7.5 of the Standard, if the OSFs / Overall Safety Requirements were defined specifically for the System, as proposed above.

Hopefully, Sub-sections 5.4.5 and 5.4.6 below will give some insight into how this could be accomplished in the development of an illustrative *Specification*, for ATM and Rail applications, respectively — the overriding need here being to demonstrate, with the appropriate degree of confidence, that the *Specification* satisfies the Requirements, given the properties of the *Application Domain*.

5.4.5 *An ATM Perspective on Arg 1.3*

Further to the comments in Sub-section 5.4.4 above, it is now recognised that Sub-section 3.4 of Fowler and Fota (2023), in simply following clause 7.5 of IEC 61508's Overall Safety Function processes at this stage, falls well short of what would be needed to satisfy Arg1.3 for the Point Merge ATM application. Whereas it derives a comprehensive list of Overall Safety Functions, and Overall Safety Requirements, there is no attempt, in Fowler and Fota (2023), to validate those requirements in terms of their completeness, correctness, consistency or sufficiency, in relation to mitigation of the associated Hazards, under all *normal*, *abnormal*, and *failure* conditions!

That said, the work described later, in Sub-section 3.6 of Fowler and Fota (2023), presents a quite detailed, and much more comprehensive and rigorous analysis, which was originally aimed at adherence to the *Design* processes specified in IEC 61508-1 clause 7.7. It is now clear that this analysis could quite easily be adapted for use at the, earlier, *Specification* level, leading to a reasonable illustration of what would need to be done in order to provide adequate Evidence in support of Arg 1.3 for a typical new ATM operational concept. For a simple operational change like Point Merge, it would also reduce greatly the amount of analysis that would then be needed at the high-level Design stage — see Sub-section 6.5, below²⁸.

5.4.6 *A Rail Perspective on Arg 1.3*

Further to the comments in Sub-sections 5.4.4 and 5.4.5 above, it is now recognised that Sub-section 4.4 of Fowler and Graebner (2023), in simply following clause 7.5 of IEC 61508's Overall Safety Function processes at this stage, also falls well short of what would be needed to satisfy Arg1.3, for the CBTC moving-block application. Whereas, it derives (from an independent, authoritative source, IEC 62290-1:2014) a very comprehensive list of Overall Safety Functions, and Overall Safety Requirements, there is no attempt, in Fowler and Graebner (2023), to validate them in terms of their completeness, correctness, consistency or sufficiency, in relation to mitigation of the associated Hazards, under all *normal*, *abnormal*, and *failure* conditions!

Fortunately, the work described in Sub-section 4.6 of Fowler and Graebner (2023) presents a quite detailed, and much more comprehensive and rigorous, analysis that was originally aimed at adherence to the Design processes specified in IEC 61508-1 clause 7.7. It is now clear that the various analysis techniques that are described in Sub-section 4.6 of Fowler and Graebner (2023) could (and should) be adapted for use *also* at the earlier, *Specification* level and, in so doing, would give a reasonable illustration of what would need to be done in order to provide adequate Evidence in support of Arg 1.3 for a typical new moving-block railway-signalling system; in so doing, it would also simplify somewhat the analysis that would then be needed at the high-level Design stage — see Sub-section 6.6, below.

²⁸ The author intends to present a revised version of the analysis in Sub-section 3.4 of Fowler and Fota (2023) in a 2027 issue of the Safety-critical Systems eJournal

6 Argument 2 — System Design

6.1 Overview

Argument 2 asserts that the high-level *Design* of the System satisfies its *Specification*, correctly and completely, under all *normal*, *abnormal*, and *failure* conditions. Whereas the *Specification* sets out the rules about what *must happen* in the *Application Domain*, the *Design* specifies the Safety Functions that *must be performed* by the System, such that that the rules are satisfied, in the *Application Domain*, under all the stated conditions.

Furthermore, in order to provide an adequate basis for Argument 3, i.e. the argument that the *implementation* of the *Design*, in the built-and-tested System, satisfies the *Design* requirements correctly and completely, the *Design* would typically be based on a high-level *Architecture* of the System, specifying:

- one or more *Logical* (rather than physical) elements — what IEC 61508-4 defines as Safety Related Systems (SRSs) — and the interactions between them; and
- the individual *Safety Functions* to be performed by each SRS.

In the case of functionally complex applications, e.g. railway signalling systems, it would usually be necessary to complement the *Logical Architecture* with separate, *Functional Models* of the system, with cross-checking between these two views of the *Design*.

6.2 Strategy

The *Strategy* for satisfying Arg 2 is to show that the high-level *Design* correctly states what is necessary and sufficient to satisfy the System *Specification*, whilst providing a necessary and sufficient set of requirements for the achievement of a tolerable level of safety in the physical Implementation of that *Design*.

Specifically, it is necessary to provide arguments and evidence which show that:

1. the *Design* for the System states correctly what is necessary and sufficient to satisfy the Overall Safety Requirements specified in the System *Specification*, under all *normal*, *abnormal*, and *failure* states of the EUC, the EUC Control System and the Operational Environment, as well as the transitions between, and recovery from, those states (**Arg 2.1**); and
2. the *Design* states correctly what is necessary and sufficient, and in enough detail and clarity, to enable the subsequent implementation of the System *Design* in the *Implementation*, to achieve the required level of safety for the *Application Domain*, for all *normal*, *abnormal*, and *failure* conditions (**Arg 2.2**).

6.3 Assurance Objectives for Arg 2

Credible arguments and evidence shall be provided to demonstrate the achievement of each of the 17 Assurance Objectives defined for Arg 2.1, in Table 8 and Table 9; and the 11 Assurance Objectives defined for Arg 2.2, in Table 10 and Table 11.

Table 8 ~ Direct Assurance Objectives for Arg 2.1

ID	What has to be Demonstrated
2.1.1	A necessary and sufficient set of Safety-related Systems (SRSs) has been identified in response to the Overall Safety Functions and Overall Safety Requirements of the System <i>Specification</i>
2.1.2	The safety requirements set out in the System <i>Specification</i> have been completely and correctly allocated and apportioned to the set of SRSs, such that the Overall Safety Requirements for each Overall Safety Function are capable of being satisfied
2.1.3	At least one <i>Safety Function</i> has been developed for each SRS, clearly and accurately reflecting the System Design's required contribution to the rule-based requirements at the Specification level <i>Note: the terms Safety Function and SRS are as defined in IEC 61508-4 — see Sub-section 1.4, above.</i>
2.1.4	Each <i>Safety Function</i> has been correctly identified as being applicable to an SRS. operating in low demand, high demand or continuous modes of operation
2.1.5	The Functional Safety Requirements specified for each <i>Safety Function</i> are sufficient to satisfy the Overall Safety Requirements allocated to its host SRS
2.1.6	The Functional Safety Requirements specified for each <i>Safety Function</i> take full account of all relevant information related to the Application Domain, at this level of the Design
2.1.7	The Functional Safety Requirements for each <i>Safety Function</i> adequately specify the required safety behaviour of the System at the <i>Design</i> level, for all normal states of the EUC, the EUC Control System and the Operational Environment, and the transitions between, and recovery from, those states
2.1.8	The Functional Safety Requirements for each <i>Safety Function</i> adequately specify the required safety behaviour of the System at the <i>Design</i> level, for all abnormal states of the EUC, the EUC Control System and the Operational Environment, and the transitions between those states, including recovery of the System to <i>normal</i> operations
2.1.9	The Functional Safety Requirements for each <i>Safety Function</i> adequately specify the required safety behaviour of the System at the <i>Design</i> level, for all failure states of the EUC, the EUC Control System and the Operational Environment, and the transitions between those states, including recovery of the System to <i>normal</i> operations
2.1.10	The Functional Safety Requirements for each <i>Safety Function</i> adequately specify the required safety attributes of the <i>Safety Function</i>
2.1.11	A Safety Integrity Requirement (SIR) has been specified for each <i>Safety Function</i> , such that the corresponding target SIR(s) in the System Specification would be met

ID	What has to be Demonstrated
2.1.12	For SRSs that implement Safety Functions of different Safety-integrity Levels, it has been shown that either: • there would be sufficient independence of implementation between these particular Safety Functions; or • the requirements applicable to the highest relevant Safety Integrity Level apply to all Safety Functions

Table 9 ~ *Backing Assurance Objectives for Arg 2.1*

ID	What has to be Demonstrated
2.1.13	The analysis methods, techniques and tools used in the derivation and verification of the Functional Safety Requirements for each Safety Function are appropriate for the attributes of those requirements
2.1.14	The analysis notations are appropriate to the Application Domain and the Safety Integrity Levels derived for each Safety Function
2.1.15	The analysis notations allow an adequate analysis of the <i>Design</i>
2.1.16	The allocation of the Safety Integrity Requirements was carried out using appropriate techniques for the combination of probabilities
2.1.17	Only suitably qualified and experienced staff have applied the analysis methods, techniques and tools

Table 10 ~ *Direct Assurance Objectives for Arg 2.2*

ID	What has to be Demonstrated
2.2.1	The System high-level <i>Design</i> safety requirements are specified in sufficient detail and clarity to allow the <i>Implementation</i> of the System to achieve the required level of safety
2.2.2	The Functional Safety Requirements for each <i>Safety Function</i> are specified explicitly as such, and are set out in such a way as to be easily distinguishable from other (non-safety) requirements
2.2.3	The System high-level <i>Design</i> safety requirements specify the necessary limiting and constraint conditions for the <i>Implementation</i> of the System high-level <i>Design</i> due to the possibility of common cause failures
2.2.4	There are no emergent properties of the System high-level <i>Design</i> that could allow non-safety functions to interfere with Safety Functions and the Data sources / flows, in the <i>Implementation</i> of the System <i>Design</i>
2.2.5	All feasible measures have been taken to avoid that necessity of the requirement for a SIL 4 Safety Function in the <i>Design</i>
2.2.7	All Functional Safety Requirements and Safety Integrity Requirements for the high-level <i>Design</i> are realistic, i.e. are capable of being implemented by known, existing technology
2.2.8	All Functional Safety Requirements and Safety Integrity Requirements for the high-level <i>Design</i> are verifiable at the <i>Implementation</i> level

Table 11 ~ Backing Assurance Objectives for Arg 2.2

ID	What has to be Demonstrated
2.2.9	The analysis methods, techniques and tools used in assessment of the suitability of the Functional Safety Requirements for each Safety Function are appropriate for the attributes of those requirements
2.2.10	The analysis methods, techniques and tools used in assessment of the suitability of the Functional Safety Requirements are appropriate to the Application Domain and the Safety Integrity Levels derived for each Safety Function
2.2.11	Only suitably qualified and experienced staff have applied the analysis methods, techniques and tools used in assessment of the suitability of the Functional Safety Requirements.

6.4 Supporting Evidence for Arg 2 from IEC 61508

Subclause 7.10.1 of IEC 61508-1 states that the “*objective*” [sic] of Phase 9 — System Safety Requirements Specification — is to, “*define the ... [SRS] safety requirements, in terms of the [SRS] safety functions ... and the [SRS] safety integrity requirements, in order to achieve the required functional safety*”.

Subclause 7.10.2.1 then adds that, “*The ... [SRS] safety requirements specification shall be derived from the allocation of safety requirements specified in [Phase 5] together with all relevant information related to the application. This information shall be made available to the ... [SRS] developer*”.

Whereas the overall intent is clear, the traceability back to the Overall Safety Requirements (see Sub-section 5.4 above) is weak, and the stated objective of achieving “*the required functional safety*” is (for complex, transport applications, at least) wholly inadequate in terms of essential rich traceability back to the top-level Application Domain Requirements (see Sub-section 5.3 above). These are a further reflection on the weaknesses of Phase 5 of IEC 61508-1, as noted in Sub-section 5.4.4 above.

Otherwise, Phase 9 of IEC 61508-1, and subclause 7.10.2.6 in particular, places great emphasis on the need for a description of the workings of the SRS, including:

- a description of all the Safety Functions, how they work together to satisfy the rule-based requirements of the Overall Safety Functions, and whether they operate in low-demand, high-demand or continuous modes of operation;
- the required performance attributes of each safety function - e.g. timing properties and (for more data-intensive applications than possibly envisaged by IEC 61508) data accuracy, latency, refresh rate, and overload tolerance;
- all interfaces²⁹ that are necessary to achieve the required functional safety;
- all relevant modes of operation of the EUC and its Control System;
- response of the SRSs to abnormal conditions that might arise in the EUC, its Control System or its Operational Environment;
- all required modes of behaviour of the SRSs — in particular, its failure behaviour and the required response in the event of such failure; and

²⁹ IEC 61508-1 also includes “operator” interfaces at this level. These have been excluded here on the ground that human operators can by definition form part of a physical SRS and should be left until the physical (or at least logical) design stage of the development lifecycle.

- the need for the safety requirements at this level to be sufficient to enable a complete and correct implementation of the high-level Design, in the physical system(s).

Indeed, much use has been made of, *inter alia*, clause 7.10 of IEC 61508-1 in framing the Assurance Objectives for Arg 2, in Sub-section 6.3 above.

6.5 An ATM Perspective on Arg 2

In relation to an ATM operational concept known as Point Merge, Sub-section 3.5 of Fowler and Fota (2023) suggests how the *Functional Safety Requirements* and *Safety Integrity Requirements*, which were derived for the three *Overall Safety Functions* in Sub-section 3.4 thereto, could be allocated to a total of 13 *Safety-related Systems*, as per clause 7.6 of IEC 61508-1³⁰.

Sub-section 3.6 of Fowler and Fota (2023) then purports to “*develop safety requirements for those Safety-related Systems, in terms of their Functional Safety Requirements ... and [Safety Integrity Requirements], in order to achieve the required functional safety under all normal, abnormal and failure conditions*”, as per clause 7.10 of IEC 61508-1.

Whereas these are both reasonable attempts to follow the related IEC 61508-1 safety processes, they also inherit two weaknesses from the Standard, revealed by the safety-argument approach, viz:

- lack of rigour in the Specification, and validation, of the *Overall Safety Functions*, as reported in Sub-sections 5.4.4 and 5.4.5 above;
- the lack of rich traceability, back to Application Domain Requirements, through the Overall Safety Requirements, as implied in the in use of the inadequate phrase “*in order to achieve the required functional safety*”.

However, as already been explained, in Sub-section 5.4.5 above, most of the scenario analysis presented in Sub-section 3.6 of Fowler and Fota (2023) would much more usefully have been carried out at the *Specification* level in order to enhance the *Overall Safety Requirements* thereof. This would have required only small changes in the terminology used, e.g. Overall Safety Requirements instead of Functional Safety Requirements, and strict adherence to the rule-based principle of requirements expression in the Specification; there would also be no need to repeat the analysis at the *Safety-related System* level since, as noted in Sub-section 3.6.2 of Fowler and Fota (2023):

“In the particular case of Point Merge operations, there are no new SRSs / Safety Functions; rather, the operational concept is based on existing ... elements of ... what IEC 61508-4 terms the EUC Control System.”

In effect, this would mean that, for Point Merge, the main high-level Design outputs would be the design of the *Precision Area Navigation* (P-RNAV) routes, which make up the Approach patterns, and the human tasks that would need to be carried out (mainly by flight crew and air traffic controllers), and when they were to be carried out, in order to satisfy the rule-based Overall Safety Requirements, under the stated conditions.

Finally, Sub-section 3.6.6 of Fowler and Fota (2023) addresses the question of how failures *internal* to the ATM System impact on the safety of Point Merge operations. It notes that

³⁰ No *Other Risk-reduction Methods* were identified for the Point Merge application described.

IEC 61508 suggests a Risk Classification Scheme (RCS) as a possible method for deriving Safety Integrity Requirements at this level but, having already cast doubt on the validity of RCSs used traditionally in ATM, it outlines a scheme based on that used on the SESAR Programme (SESAR 2018a), which resolves most, if not all, of those doubts. Although lacking in specific detail regarding failure-assessment processes and outputs, it does at least provide insights into the use of “*appropriate techniques for the combination of probabilities*” as required Assurance Objective #2.1.16, in Table 9 above.

6.6 A Rail Perspective on Arg 2

Sub-section 4.5 of Fowler and Graebner (2023) suggests how the eight *Overall Safety Functions*, derived in Sub-section 4.4 thereto, could be allocated to nine *Safety-related Systems* and a number of associated *Other Risk-reduction Methods*³¹. For the purposes of that exercise, the SRSs were not pure invention; rather, they had been adapted directly from the highest-level functional elements set out in the *Urban Guided Transport Management and Command/Control Systems* rail standard, IEC 62290-1.

Sub-section 4.6.3 of Fowler and Graebner (2023) then derives, and describes, a total of 30 illustrative Safety Functions covering the nine SRSs. Again, this task was made easier by adapting the Safety Functions, and their descriptions, from IEC 62290-1. As noted therein, this initial set of Safety Functions covered only those necessary for *normal* operational conditions. Following the same strategy, it was noted that some 350 Functional Safety Requirements would be needed to cover the 30 Safety Functions; Appendix B to Fowler and Graebner (2023) therefore presented, as examples, just the 41 requirements that would apply to the three SRSs needed to support the Overall Safety Function “*Establish & Protect a Safe Route for each Train Movement*”.

Recognising that the above examples had been derived solely hierarchically, Sub-section 4.6.4 of Fowler and Graebner (2023) sought to show explicitly:

- how the functions interact with each other, and with the elements of the wider CBTC system and the EUC Control System;
- the information needed by, and produced by, the SRSs and Safety Functions;
- the system states and sequence of events, during a typical “day-in-the-life” of a train; and
- that the requirements constitute a complete, correct and coherent set.

It is noted therein that there is a wide range of techniques for addressing these issues and gives worked examples of the use of:

- State-transition Models of the various operational states of the railway;
- Operational Scenario Analysis;
- Sequence Diagrams for specific scenarios; and
- Activity Diagrams for capturing the interactions between Safety Functions and the Data involved.

Sub-section 4.6.5 of Fowler and Graebner (2023) outlines an equivalent approach for *abnormal* operational conditions, which stem mainly from:

³¹ For the three most important OSFs, and the five most important SRSs, the relationships involved in the allocations were one-to-many, making the subsequent derivation of requirements for Safety Functions somewhat more complex than would otherwise have been the case!

- hazardous events in the operational environment that are not encountered on a day-to-day basis; and
- failure events within the EUC Control System but *outside* the scope of the CBTC system itself, e.g. a failure of a train's traction-control system.

It is noted therein that, in either case, the following need to be addressed:

- what effect the event would have on the continuing functioning of the CBTC system and the consequences for the safe operation of the railway; and
- what actions would need to be taken to mitigate the consequences of the event, and how the functionality of the CBTC system (existing or additional) could be used to support such actions.

Finally, Sub-section 4.6.6 of Fowler and Graebner (2023) addresses the question of how failures *internal* to the CBTC System impact on the safety of railway operations. It notes that, as such analysis is “*covered comprehensively in existing rail safety standards, including EN 50126-1*”, Sub-section 4.6.6 is limited to addressing key principles relating specifically to the IEC 61508-1 viewpoint. However, although lacking in specific detail regarding failure-assessment processes and outputs, it does at least provide insights into the use of “*appropriate techniques for the combination of probabilities*” as required by Assurance Objective #2.1.16, in Table 9 above.

Overall, the aim of Fowler and Graebner (2023) was to show what the safety assessment of a railway signalling system would look like, if based on IEC 61508-1; given that proviso, Sub-sections 4.5 and 4.6 of that article do a reasonably good job of illustrating what the safety assurance model in Figure 1 refers to as Assurance Activities and Activity Outputs. What it lacks (as does IEC 61508-1 largely) is much detail on the subsequent processes involved in Output Analysis, which are so vital to the submission of Evidence to satisfy the Assurance Objectives and thence to support the related Safety Argument.

7 Conclusions

The 1970s through 1990s was a bad period for international system safety, across a variety of safety-critical industries. Individual reaction to serious accidents led gradually to a worldwide understanding of the need for the development of a holistic approach to safety management and what has become recognisable as typical safety-management systems. What also emerged, in 1990, was an extremely important finding of the Cullen inquiry into the 1998 Piper Alpha oil platform disaster, which “*recommend that operators be legally required to produce a comprehensive Safety Case as a means of assuring themselves that their operations are acceptably safe*” (Cullen 1990).

In parallel with these activities, and in the face of emerging new technologies, the IEC started work on a generic standard for the assessment of Programmable Electronic Systems (PES) used in safety-related applications. Over the period 1998 to 2000, following earlier releases of draft material, the first full editions of the seven parts of IEC 61508 were published. IEC 61508 was, and *still* is, widely accepted as the “gold standard” in Functional Safety, i.e. in the safety assessment of Safety Related Systems whose primary purpose is to provide mitigation of operating hazards that are inherent in many disparate safety-related industries sectors. Its stated purpose has always been to provide a solid, comprehensive basis for adaptation to a very wide range of applications — in that general sense, it has

undoubtedly been successful, though, at a more detailed level, the completeness and consistency of some of the adaptations of the Standard, released with regard to the original, might be called into question — EUROCONTROL’s System Safety Assessment “Methodology” and the CENELEC railway RAMS standard EN 50126:1999 could be considered to be cases in point.

Fast-forwarding to today we see, in the European ATM and Rail sectors, respectively:

- the appearance of a EUROCONTROL produced³², and recently updated, safety assessment manual, with the somewhat vague title “*Safety Reference Material*”, with accompanying “*Guidance*” for applying it; and
- major updates to the CENELEC standards EN 50126:2017, EN 50716:2023 (formerly EN 50128) and EN 50129:2026.

The SESAR (2018a) “*Safety Reference Material*” makes no claim to be compliant with IEC 61508 but, to the firsthand knowledge of the author of this article, it does at least adhere to the fundamental principles of IEC 61508-1, outlined herein.

EN 50126:2017, on the other hand, is quite specific on this point and *implies* that the three CENELEC standards together do comply with IEC 61508 — a huge claim. Whereas the latest versions of the three CENELEC standards do seem to bear close resemblance to IEC 61508³³, they do include what might be viewed as a potential logical inconsistency in that, apparently for reasons of compliance with Regulation (EU) No 402/2013, “*Common Safety Method for Risk Evaluation and Assessment*”, they use the variable concept of a “*System Under Consideration*” as the starting point for a safety assessment, rather than the very specific IEC 61508 concept of Equipment Under Control.

Over many decades, Professor Nancy Leveson of MIT has noticed that “... *system safety engineering has increasingly drifted toward using reliability engineering techniques and away from classic system safety approaches* ...” and this has resulted in the fact that accidents she studied had involved software that did exactly what it was designed to do, but for which the designers had misunderstood what *behaviours* were required from the software in order to achieve an acceptable level of safety — in other words, the software reliably performed the *wrong* functions. This exactly what can happen if, when a new system is introduced into an existing operation, that system (instead of the overall operation) becomes the main (or, worse still, sole) focus of the safety assessment process — what Jackson (1995) refers to as, and warns about, an imbalance in “*the competing demands and attractions of the world and the machine*” during the development process.

In any case, what is generally well understood is that all the aforementioned standards are primarily process-based, and whilst specifying the *assurance activities* to be followed in system safety assessment, do not guarantee that the activity outputs would provide sufficient *evidence* for a *safety case* that showed that the *system under consideration* would enable an acceptable level of safety for the overall operation. For that reason, it was decided that a more productive approach for this article would be to develop and to present a rigorous and independent Safety Argument structure, against which the effectiveness of system safety assessment standards, such as the SESAR Safety Reference Material or the CENELEC series, (and including IEC 61508-1 itself!) could be judged individually, rather than comparing one set of standards with another.

³² With others, and on behalf of the European Commission’s Single European Sky ATM Research & Development (SESAR) Programme.

³³ Certainly, a *much* closer resemblance than earlier versions see Fowler (2015).

Such a Safety Argument structure is presented herein. Although it is based on the fundamental principles underlying IEC 61508, and generally aligned with the IEC 61508-1 safety lifecycle, it is otherwise largely independent of the provisions of that Standard.

The key, and to some extent unique, features of the Safety Argument framework are as follows:

Firstly, the Argument comprises five top-level claims, covering the whole of a typical system lifecycle, and is concerned entirely with *Functional Safety*, i.e. safety relating to an inherently hazardous “Equipment Under Control”, which, for the purposes of all four articles in this series, has been interpreted as the flow of road, rail or air traffic, in the area of interest and which depends on the *correct functioning* of Safety-related Systems, and/or other means of reducing the associated inherent risks to an acceptable level.

Secondly, the first two claims, (Argument 1 — System Specification, and Argument 2 — System High-level Design) are based on a rigorous Requirements-engineering model, adapted from the seminal work of Dr Michael A Jackson (Jackson 1995) and underpinned by the principles of *Rich Traceability*. The adaptation of Jackson’s model makes many vital contributions to Argument 1, in particular, by introducing the concept of a *Specification* that:

- captures the properties shared by the *System* and its *Application Domain* (the latter comprising the EUC, the EUC Control System and the Operational Environment);
- states only what the System has to *make happen* (as opposed what the System has to *do*) in the Application Domain; hence, for the transport sector at least, the Specification expresses its safety requirements as *Rules* to be applied in the Application Domain (i.e. rules of the road, sea, rail or air), *rather than* safety functions to be provided by a *safety-related system*; and
- has to be shown to be necessary and *sufficient in itself* to satisfy the safety Requirements, i.e. the *Safety Targets*, of the Application Domain.

It also forces Argument 1 to be conditional on complete and correct knowledge of the related properties (actual or assumed) of the Application Domain.

A review is included of three cases from the transport sector, in which seeming ignorance of the principles of Jackson (1995) led to catastrophic failures; this has underlined the importance of that work to the framing of a sound Safety Argument from the outset.

Thirdly, the Argument / Evidence structure is founded on a Safety Assurance Model in which the lowest level of decomposition of the Safety Argument, and the highest level in the Safety Assurance Model, are sets of Assurance *Objectives*, which prescribe exactly what has to be *demonstrated* by the Evidence, i.e. by the outputs of the Assurance Activities. These Assurance Objectives are based quite heavily on a system-level interpretation of a very comprehensive set of *Regulatory Objectives for Software Safety Assurance in ATS Equipment*, published by the UK Civil Aviation Authority (UK CAA 2019).

An assessment of the provisions of IEC 61508-1, within the scope of the three predecessor articles, then showed that:

- there is a significant deficiency in Phase 4 of IEC 61508-1 (Overall Safety Requirements), in relation to its *Specification* equivalent; whereas the Standard does mandate the derivation of technology-independent Overall Safety Requirements for each Overall Safety Function, the treatment is somewhat superficial, and there are no explicit requirements to validate the completeness, correctness or sufficiency of the Overall Safety Requirements in relation to the satisfaction of the Application Domain Requirements;

- Phase 5 of IEC 61508-1 (High-level Design) rightly makes demands regarding the rigour and completeness of the Design information; however, it lacks explicit requirements for demonstrating, *via the Specification*, that the high-level Design satisfies the top-level, safety requirements of the Application Domain; this would not be an adequate situation for many safety-related systems in the transport sector, due to their much greater complexity at the Design level compared with the process industries wherein lie the origins of IEC 61508.

Otherwise, as shown previously in the assessment of the example applications for the ATM sector in Fowler and Fota (2023) and for the Rail sector in Fowler and Graebner (2023), IEC 61508-1 makes a considerable contribution to the definition of Safety Assurance *Activities* but, being a generic, process-based standard, would not satisfy the majority of the Assurance Objectives derived herein.

It is concluded, therefore, that as far as IEC 61508-1 is concerned, the Safety Argument would not be satisfied purely by adherence to the Standard, although the Standard does provide most of what would be required in the way of specifying the Safety Assurance Activities needed to produce sufficient Evidence to support the Safety Argument.

Given that, the question of whether or not “... [EN 50126], together with the requirements of other suitable standards, is sufficient to ensure that additional compliance to IEC 61508 does not need to be demonstrated” seems to be not only difficult to justify but also to be largely irrelevant. Therefore, far more important questions would be to what degree, and extent, would compliance with those railway safety standards satisfy the Safety Argument presented herein — and, if not, why not?

Disclaimers

Any opinions expressed herein are those of the author, not necessarily those of any of his past or present employers or clients.

Whilst every effort has been made to provide a comprehensive set of generic Assurance Objectives, in the context of the article, they should be understood as being illustrative and non-exhaustive, rather than as being definitive for any particular application.

Correspondence Address

Corresponding e-mail address: derek-fowler@btconnect.com.

Acknowledgments

The author wishes to acknowledge the help, support, understanding of IEC 61508, and experience of aviation and rail safety, provided by his long-standing colleague Ron Pierce, without which this series of articles would not have come to fruition.

The copyright holder of the quotations from published standards used for illustration in this article is the International Electrotechnical Commission, Geneva. The primary copyright holder for EN 50126, and associated standards, is CENELEC (the European Committee for Electrotechnical Standardization), Brussels. The SESAR 3 Joint Undertaking (SESAR JU) holds the direct copyright ownership of SESAR material; it mandates that while technical project results are technically owned by the participating industry beneficiaries, they are licensed back to the SESAR JU.

References

- BFU. (2004). Bundestelle für Flugunfalluntersuchung (German Federal Bureau of Aircraft Accident Investigation) Investigation Report AX001-1-2/02, May 2004. Available from: <https://skybrary.aero/sites/default/files/bookshelf/414.pdf>. Accessed 11th September 2026.
- CLC/TR 50451. (2007). *Railway applications – Systematic allocation of safety integrity requirements*. CENELEC (European Committee for Electrotechnical Standardization), Brussels.
- CLC/TR 50506. (2007). *Railway Applications – Communication, signalling and processing systems – application guide for EN 50129*. CENELEC (European Committee for Electrotechnical Standardization), Brussels.
- Cogent Skills. (2026). *The Bhopal tragedy and its impact on process safety*. Cogent Skills. <https://cogentskills.com/news/the-bhopal-tragedy-and-its-impact-on-process-safety>. Accessed 17th July 2026.
- Cullen W. D. (1990). *The Public Inquiry into the Piper Alpha Disaster*. HMSO Publications Centre, ISBN 0–10–113102-X.
- Dick J. (2002). *Rich Traceability*. ResearchGate, January 2002. Available from: https://www.researchgate.net/publication/239560722_Rich_Traceability. Accessed 17th July 2026.
- Dowson M. (1997). *The ARIANE 5 Software Failure*. In ACM SIGSOFT Software Engineering Notes Volume 22, No. 2, March 1997. Available from: <https://dl.acm.org/doi/pdf/10.1145/251880.251992>. Accessed 17th July 2026.
- (EU) No 402/2013. (2013). *Commission Implementing Regulation (EU) No 402/2013 of 30 April 2013 on the common safety method for risk evaluation and assessment and repealing Regulation (EC) No 352/2009*. European Commission, Brussels.
- EN 50126-1. (2017). *Railway applications – the specification and demonstration of reliability, availability, maintainability and safety (RAMS), Part 1: Basic requirements and generic process*. EN 50126-1:2017. CENELEC (European Committee for Electrotechnical Standardization), Brussels.
- EN 50128. (2020). *Railway applications – Communication, signalling and processing systems – Software for railway control and protection systems*. EN 50128:2020. CENELEC (European Committee for Electrotechnical Standardization), Brussels.
- EN 50129. (2003). *Railway applications - Communication, signalling and processing systems - Safety related electronic systems for signalling*. EN 50129:2003, CENELEC (European Committee for Electrotechnical Standardization), Brussels.
- EN 50716. (2023). *Railway Applications — Requirements for software development*. EN 50716, 1st Edition, 2023. CENELEC (European Committee for Electrotechnical Standardization), Brussels.
- EUROCONTROL. (2006a). *Air Navigation System Safety Assessment Methodology*. EUROCONTROL SAF.ET1.ST03.1000-MAN-01, 3 October 2006. Available at <https://skybrary.aero/sam-toolkit>. Accessed 13th July 2026.

- EUROCONTROL. (2006b). *Safety Case Development Manual*. EUROCONTROL DAP/SSH/091, Edition 2.1, 13 October 2006. Available at <https://skybrary.aero/sites/default/files/bookshelf/33520.pdf>. Accessed 17th July 2026.
- EUROCONTROL. (2011). *ACAS II Post-implementation Safety Case*. EUROCONTROL ALDA Reference 11/03/28-15, Version 2.3, 25 November 2011. Available at <https://skybrary.aero/sites/default/files/bookshelf/4967.pdf>. Accessed 17th July 2026.
- EUROCONTROL. (2022). *ACAS II Bulletin – Near collision over Yaizu*. Available at <https://skybrary.aero/sites/default/files/bookshelf/32673.pdf>. Accessed 17th July 2026.
- Fowler D. (2015). *Functional Safety by Design – Magic or Logic?* In Proceedings of the 23rd Safety-Critical Systems Symposium, Bristol, UK.
- Fowler D. (2022). *IEC 61508 Viewpoint on System Safety in the Transport Sector: Part 1 — An Overview of IEC 61508*. Safety-Critical Systems eJournal, Vol. 1, Iss. 2. Available at <https://scsc.uk/r176.3:1>. Accessed 10th September 2026.
- Fowler D, and Fota O. N. (2023). *Safety Assessment of Point Merge Operations in Terminal Airspace — an IEC 61508 Viewpoint*. Safety Critical Club eJournal, Vol. 2, Iss. 1. Available at <https://scsc.uk/r183.3:1#page=25>. Accessed 10th September 2026.
- Fowler D, and Graebner A. (2023). *An IEC 61508 Viewpoint on the Safety Assessment of Railway Control Systems*. Safety Critical Club eJournal, Vol. 2, Iss. 2. Available at <https://scsc.uk/r1858>. Accessed 10th September 2026.
- Fowler D, Perrin E, and Pierce R. (2009). *2020 Foresight — a Systems-engineering Approach to Assessing the Safety of the SESAR Operational Concept*. Proceedings of the 8th USA/Europe Air Traffic Management Research and Development Seminar, Napa, USA
- Fowler D, and Pierce R. (2012). *A Safety Engineering Perspective*. In: Cogan B. (editor) *Systems Engineering: Practice and Theory*. IntechOpen. London.
- Fowler D, and Tiemeyer B. (2002). *Reduced Vertical Separation in European Airspace – A Case Study in Effective Safety Management*. Proceedings of the 20th International System Safety Conference, Denver, USA August 2002.
- Hammond J, Rawlings R, and Hall A. (2001). *Will it Work?* In the Proceedings of the 5th IEEE International Symposium on Requirements Engineering, Toronto, Canada, 2001, ISBN 0-7695-1125-2
- ICAO. (2006). *Aircraft Operations Volume I, Flight Procedures*. International Civil Aviation Organisation Doc 8168 OPS/611, 5th Edition, 2006.
- ICAO. (2007). *Procedures for Air Navigation Services*. International Civil Aviation Organisation Doc 4444, 15th Edition, 2007.
- ICO. (2022). *Explaining decisions made with AI – Annexe 5: Argument-based assurance cases*. 17th October 2022. UK Information Commissioner's Office. Available at <https://ico.org.uk/for-organisations/uk-gdpr-guidance-and-resources/artificial-intelligence/explaining-decisions-made-with-artificial-intelligence/annexe-5-argument-based-assurance-cases/>. Accessed 2nd June 2026.

- IEC 61508. (2010). *Functional safety of electrical/electronic/programmable electronic safety-related systems*. IEC 61508, in 7 parts³⁴, 2nd Edition, 2010. International Electrotechnical Commission, Geneva.
- IEC 62290-1. (2014). *Railway applications – Urban Guided Transport Management and Command/Control Systems – Part 1: System Principles and Fundamental Concepts*. IEC 62290-1:2014, 2nd Edition, 2014. International Electrotechnical Commission, Geneva.
- IRJ. (2014). *London Underground and Bombardier abandon Tube signalling contract*. International Railway Journal, 3 January 2014. <https://www.railjournal.com/financial/london-underground-and-bombardier-abandon-tube-signalling-contract/>. Accessed 17th July 2026.
- IRJ. (2015). *Thales wins £760m London Underground re-signalling contract*. International Railway Journal, 3 August 2015. https://www.railway-technology.com/news/newsthales-wins-760m-london-underground-re-signalling-contract-4638341/?utm_source=chatgpt.com&cf-view. Accessed 27 October 2025.
- Jackson M. A. (1995). *The World and the Machine*. In the Proceedings of the 17th IEEE International Conference on Software Engineering (ICSE), Seattle, WA, 1995, ISBN 0-89791-708-1
- Leveson N. G. (2001). *The Role of Software in Recent Aerospace Accidents*. In the Proceedings of the 19th International System Safety Conference, Huntsville, Alabama, USA, 10-14 September 2001. Available at <http://sunnyday.mit.edu/accidents/issc01.pdf>. Accessed 17th July 2026.
- Leveson N. G. (2020). *Are You Sure Your Software Will Not Kill Anyone?* Communications of the Association of Computing Machinery (ACM), 1st February 2020. Available at: <https://cacm.acm.org/opinion/are-you-sure-your-software-will-not-kill-anyone/>. Accessed 12th July 2026.
- Lions J-L. (1996). *Ariane 5 — Flight 501 Failure*. Report by the Inquiry Board, Paris, France 19 July 1996. Available at: <https://esamultimedia.esa.int/docs/esa-x-1819eng.pdf>. Accessed 17th July 2026.
- London Assembly. (2016). *Transport for London's Signal Failure*. The London Assembly's Budget and Performance Committee Report dated March 2016. Available at: https://www.london.gov.uk/sites/default/files/transport_for_londons_signal_failure.pdf. Accessed 17th July 2026.
- Lucic I. (2015). *Risk and Safety in Engineering Processes*. Cambridge Scholars Publishing, January 2015. ISBN (13): 978-1-4438-7077-1.
- Mannan M. S, West H. H, Krishna K, Aldeeb A. A, Keren N, Saraf S. R, Liu Y, and Gentile M. (2005). *The legacy of Bhopal: The impact over the last 20 years and future direction*. Journal of Loss Prevention in the Process Industries. Volume 18, Issues 4–6, 2005, Pages 218-224. Available at: <https://doi.org/10.1016/j.jlp.2005.06.037>. Accessed 24th June 2026.
- ORR. (2024). *Common Safety Method for Risk Evaluation and Assessment*. UK Office of Rail and Road, 16th December 2024.

³⁴ Parts are typically denoted with hyphen and part number, e.g. IEC 61508-3:2010 is Part 3, dealing with software development requirements.

- Redmill F. (1999). *An Introduction to the Safety Standard IEC 61508*. Journal of the System Safety Society, Volume 35, No. 1, First Quarter 1999. The International System Safety Society, Britton, SD.
- SAE ARP4754A. (1996). *Recommended Practice, Guidelines for Development of Civil Aircraft and Systems*. ARP4754A. SAE International, Warrendale, PA. Available at <https://doi.org/10.4271/ARP4754A>. Accessed 13th July 2026.
- SAE ARP4761. (1996). *Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment*. ARP 4761. SAE International, Warrendale, PA. Available at <https://doi.org/10.4271/ARP4761>. Accessed 13th July 2026.
- SESAR. (2018a). *SESAR Safety Reference Material*. Edition 00.04.01, 14th December 2018. SESAR Joint Undertaking. Available at [https://www.sesarju.eu/sites/default/files/documents/transversal/SESAR2020%20Safety%20Reference%20Material%20Ed%2000_04_01_1%20\(1_0\).pdf](https://www.sesarju.eu/sites/default/files/documents/transversal/SESAR2020%20Safety%20Reference%20Material%20Ed%2000_04_01_1%20(1_0).pdf). Accessed 17th July 2026.
- SESAR. (2018b). *Guidance to Apply SESAR Safety Reference Material*. Edition 00.03.01, 14th December 2018. SESAR Joint Undertaking. <https://www.sesarju.eu/sites/default/files/documents/transversal/SESAR%202020%20-%20Guidance%20to%20Apply%20the%20SESAR2020%20Safety%20Reference%20Material.pdf>. Accessed 17th July 2026.
- Thomas S. M, and Fowler D. (2013). *Safety Case for the Airborne Collision Avoidance System*. In: Dale C, and Anderson T. (editors) *Assuring the Safety of Systems*, Proceedings of the Twenty-first Safety-Critical Systems Symposium, Bristol, UK, 5-7th February 2013.
- UK CAA. (2019). *Regulatory Objectives for Software Safety Assurance in ATS Equipment*. CAP 670 Part B, Section 3, SW01. UK Civil Aviation Authority, Third Issue, Amendment 1/2019, 1 June 2019.
- UKMoD. (2016). *Journey to ALARP*. UK Ministry of Defence. June 2016. Available at: <https://test.asems.mod.uk/sites/default/files/documents/Journey%20to%20ALARP.pdf>. Accessed 17th July 2026.
- Wiley R. J, Crowl D. A, and Lepkowski W. (2005). *The Bhopal tragedy: its influence on process and community safety as practiced in the United States*. Journal of Loss Prevention in the Process Industries, Volume 18, Issues 4–6, 2005, Pages 365-374. Available at: <https://doi.org/10.1016/j.jlp.2005.06.019>. Accessed 4th July 2026.

Appendix A. Good Practice in Safety Arguments & Evidence

The following principles (adapted from EUROCONTROL (2006)) are recommended to be followed in the development of an argument and evidence structures.

A.1 Decomposition of Safety Arguments

The decomposition of a top-level Claim into lower-level Arguments provides the essential links between the top-level Claim and the wealth of Evidence needed to show that the overall Claim is valid. In performing this decomposition, it is essential that:

- each Claim in the Argument structure is expressed as a simple predicate, i.e. a statement that can be only true or false;
- the Argument structure does not contain any indirect or inconclusive Claims;
- the means by which a Claim is (or is intended to be) shown to be true is not included in the Claim statement; for example, “*System testing shows that system requirements have been satisfied*” pre-supposes (incorrectly for most high-integrity systems!) that testing would be an adequate means of showing that the system requirements have been satisfied;
- the Arguments at every level of decomposition are necessary and sufficient to show that the parent Claim or Argument is true;
- a valid counterargument, which would negate the parent Claim or Argument, does not exist; for example, a fatal counterclaim to the statement (two bullets) above would be simply “*Testing is an inadequate means of showing that the system requirements have been satisfied*”;
- where the rationale for decomposition of a Claim or Argument into lower-level Arguments is not self-evident, it is explained by supporting text, i.e. a Strategy;
- the number of levels of decomposition is appropriate to the complexity of the Safety Case and/or supporting Evidence;
- each branch of the Safety Argument structure is terminated in supporting Evidence;
- a Claim or Argument may be satisfied by either a further level of decomposition or by supporting Evidence, but not both at the same level; and
- there is a clear distinction between, and correct use of, Direct (product-based) and Backing (process-based) Arguments; see section A.3 below.

A.2 “Rules of Evidence”

The corresponding essential rules of Evidence are as follows:

- Evidence must be presented only to the degree and extent necessary to support the related Claim;

Note: the issue here is that, in the context of an Argument-based approach, any “Evidence” which is unrelated to a part of that Argument is not only of no value but could also serve as a distraction from those aspects of the Safety Case that are relevant. On the other hand, any Evidence which actually undermines the validity of an Argument must

not be ignored. The existence of such Evidence must be acknowledged and explained fully in the Safety Case.

- multiple sources of Evidence under a single Claim should not be used as a substitute for further decomposition of the Argument, except where one piece of Evidence is Direct and the other is Backing;
- Evidence must be clear, conclusive and, wherever possible, objective;

***Note:** It is vital to the integrity of the Safety Case that the Evidence be presented in such a way that is clear to the reader that the Evidence does actually show the related Argument to be true, “beyond all reasonable doubt”.*

It is bad practice to present an element of a structured Argument and then refer to a mass of information as “Evidence” to substantiate the Argument.

Where Evidence is contained in appendices or external documents, a summary justifying the adequacy of the Evidence should be presented (in the Safety Case) along with the associated Argument. It is not sufficient to merely reference the Evidence with statements such as “Evidence to support the Argument is presented in ...”.

- the type of Evidence — from safety analysis, design, simulation, test, previous usage, standards compliance, etc. — must be appropriate to the Argument and required Safety Integrity (or Assurance) Level;
- “expert” opinion must not be used as Direct Evidence, unless its use is clearly justified in the Safety Case (see, however, below for the relevance of “competence” in Backing assurance);
- the rigour of the Evidence must be appropriate to the required Safety Integrity (or Assurance) Level;
- Evidence must actually relate to the correct configuration of the system under consideration; and
- there must be a clear distinction between, and correct use of, Direct (product-based) and Backing (process-based) Evidence; see section A.3 below.

A.3 Direct versus Backing Assurance

In constructing a Safety Argument, the following two main types of argument (and supporting evidence) should be presented:

- that which shows that a particular objective has been achieved, i.e. a higher-level Argument or Claim is satisfied; this is referred to as *Direct* assurance;
- that which shows that the Direct assurance is trustworthy, i.e. that it can be relied upon; this is referred to as *Backing* assurance.

Direct assurance may be thought of as that assurance which relies directly on the observable properties of a product, i.e. the output of a process, to show that the product satisfies the requirements placed upon it.

Backing assurance, on the other hand, is obtained from the properties of the processes by which Direct assurance was obtained, and shows that those processes, tools and techniques, human resources etc were appropriate, adequate and properly deployed. In general, Backing assurance should include:

- the suitability of the processes, tools and techniques etc that are used to generate the evidence obtained from the safety assessment;
- the correct application of those processes, tools and techniques; and
- the competence of the personnel applying those processes, tools and techniques.

The general rules for the use of Direct and Backing assurance are as follows:

- Direct assurance should always be supported by Backing assurance, as necessary to demonstrate the *trustworthiness* of the Direct assurance, as appropriate to the required Safety Integrity Level; and
- Backing assurance should *not* be used other than to demonstrate the *trustworthiness* of the Direct assurance.