Moving Towards Goal-Based Safety Management

Dr Holger Becht
Head of Signalling Systems Australia
RAMS Signalling Business Unit, Ansaldo STS
PO Box 1168, Eagle Farm 4009, Queensland

holger@asssc.org

Abstract

In virtually all safety-critical industries the operators of systems have to demonstrate a systematic and thorough consideration of safety. This is generally done through the application of safety standards as part of the development of safety critical systems.

Many safety assurance standards (like EN50126 (1999), IEC 61508 (1995), DEF (Aust) 5679) (1998) are very prescriptive. They require specific techniques, approaches or measures to be applied to achieve the safety objective without allowing the users to select a suite of techniques and measures best suited for their application and development environment. The application of prescriptive techniques can work well for some systems but can be a hindrance for emerging technologies.

There has therefore been an increasing trend in many industries to demonstrate safety by assuring certain goals have been achieved, rather than simply following prescriptive standards.

Goal-based safety standards are now a reality and applied in the medical industry and defence. This paper will describe the pros and cons of prescriptive and goal-based standards, and make recommendations for the evolution of future safety standards.

Keywords: Safety Goal-Based Standards, Safety Management

1 Introduction

In this paper we look at what benefits goal-based standards can provide to and if goal-based safety cases could be a valuable tool for reasoning about safety. We discuss opportunities and challenges for the development and use of goal-based safety cases. Finally we discuss the future of safety standards and investigate how this can become a reality for system safety management.

The structure of the paper is outlined as follows.

1. Why we need goal-based standards
2. What goal-based standards exist
3. Generic goal structures
4. Generic safety management goals
5. Generic safety development assurance goals
6. Generic sets of goals
7. The evidence required for assurance
8. The impact on industry safety standards

2 Background

The term "Assurance" inherently means a positive declaration intended to give confidence. It is a subjective determination of the strength of an inference. Safety assurance is the determination of the confidence that can be placed in the safety of a system. Assurance is a property of an argument's conclusion and is based upon:

1. the likelihood that the claims are true (i.e. the assurance of the claims); and
2. the extent to which the claims entail the conclusion.

Safety Assurance is therefore a qualitative statement expressing the degree of confidence that a safety claim is true. The overall assurance of a system is equal to the assurance of the top-level goal.

A Safety Case is the primary means of communicating the goals, safety requirements, safety management environment and argument for assurance of critical systems. More specifically a safety case is a documented body of evidence that provides a convincing and valid argument that a system is adequately safe for a given application in a given environment.

Although safety cases are generally accepted, there are different ways of constructing an argument and providing the supporting evidence. The three main approaches can be characterised as shown in Figure 1.

1. Assurance via a set of evidence supported claims about the system's safety behaviour.
2. The use of accepted industry "good" practices and guidelines.
3. An investigation of known potential vulnerabilities of the system.

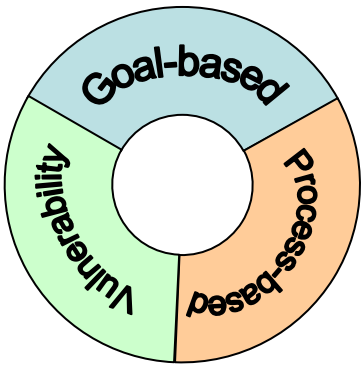

Figure 1: Safety case approaches

The first approach is goal-based – where specific safety goals for the systems are supported by arguments and evidence. The second approach is based on demonstrating compliance to a known industry accepted good practice (generally captured in a process-based safety standard). The final approach is a vulnerability-based argument where it is demonstrated that potential vulnerabilities within a system do not constitute a problem – this is essentially a "bottom-up" approach as opposed to the "top-down" approach used in goal-based methods.

These approaches are not mutually exclusive, and a combination can be used to support a safety argument, especially where the system consists of both off-the-shelf components of unknown pedigree and application-specific components.

In the past, safety arguments tended to be implicit and process-based. Compliance to accepted good practice was deemed to imply adequate safety; this is the general approach applied for most industries where compliance to standards is considered to imply adequate safety. This compliance approach works well in stable environments where good practice is supported by extensive experience, like railway signalling. However with fast moving, emerging technologies, a more pragmatic approach is required that can accommodate change and alternative strategies to achieve the same safety objective. This is why goal-based approaches are being advocated, particularly for systems with novel components and developmental systems.

3 Why Goal-Based Standards?

Historically many safety process standards have been prescriptive (i.e. tell people what to do) and/or proscriptive (i.e. tell people what to avoid doing). In contrast, goal-based standards tell people what they need to achieve (and allow alternative means to achieve this). The goal-based approach is a requirements based analysis and at a very high level, the goals are:

1. to establish safety requirements;
2. to design the system in compliance with the safety requirements; and
3. to show that the safety requirements have been fulfilled.

For example, in a goal-based approach there could be an goal to "Demonstrate completeness of the safety requirements". In "prescriptive standard" the specific means of achieving compliance is mandated; "You shall perform a Functional Failure Analysis and Accident Sequence Analysis".

Prescriptive process-based standards, like EN50128 (2001), IEC61508 (1995), DO-178B (1992), encode the good engineering practice at the time that they are written and rapidly become deficient as good practice is continuously changing with evolving technologies. In fact it is quite probable that prescriptive process eventually prevent the service provider from adopting current industry good practice.

Furthermore, technology changes rapidly and many projects find that cutting edge technology at the beginning of a project can be out-dated by the time it goes into service. The problem is that standards change relatively slowly taking up to 10 years to be updated and released. This means that prescriptive standards will always be behind the technology curve.

Consequently there are clear benefits in adopting a goal-based approach as it gives greater freedom in developing technical solutions and accommodating different technical solutions. In order to adopt a goal-based approach, it is necessary to provide a coherent and convincing safety justification.

A goal-based approach can be applied at any level from the top-level system downwards. It is important that there are clear links between the top-level goals and the sub-goals. At each level, the acceptance authority requires explicit safety goals, convincing arguments to justify the goals are met, and adequate evidence to support the arguments. In practice the rigour of the arguments and the amount of evidence will depend on the safety significance of the individual system functions.

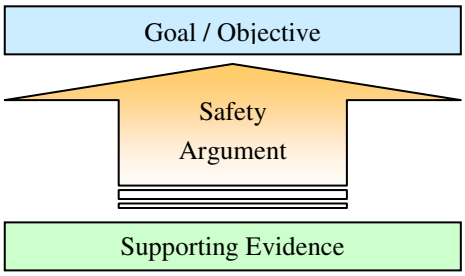

Figure 2: Goal-based Argument

The advantages, or opportunities, offered by a goal-based approach bring some attendant challenges, including:

1. Agreeing on appropriate means, and depth of evidence, for demonstrating safety, especially with emerging technology;
2. Contracting for a safety program where the set of safety activities and required evidence may not be determined "up front".

It will also be challenging for certifying bodies to certify products to a goal-based standard. With prescriptive standards this is a relative mechanical process. The certifier would assess a product by using the prescriptive requirements in that standard as a checklist to confirm compliance. With goal-based standards this is not possible and there is much more responsibility placed on the certifier who will need to make a subjective judgement instead of an objective one. Certifiers in turn will most likely shift this responsibility onto the Independent Safety Assessor to make the judgement that a specific product or system is safe and fit-for-purpose.

This means that in order for goal-based standard to be effective some of the inherent subjectivity of this approach needs to be reduced to simplify the acceptance and certification process.

4 Goal-Based Standards

Despite the differences in detail, goal-based approaches are now being adopted in standards with the key premise that they are not to be technology specific.

The UK Civil Aviation Authority software safety assurance standard, CAA SW01 (2002) identifies a standard set of top-level goals for software based systems which are generic (e.g. specification is valid, specification is correctly implemented, etc.).

The software part of Def Stan 00-56 (2007) requires goal-based safety justification and explicit safety arguments to support the safety claims made. Def Stan 00-56 (2007) may have taken the goal-based approach too far in an attempt to be completely flexible. The standard places the entire onus on the service provider to develop the system as they please and provide justification that the system is safe. It is clear from this standard that some structure and minimal processes need to be prescribed. In reality we see both approaches working in parallel. The Yellow Book is one of the few standards that provides high level goals and suggests several process-based standards to achieve each goal.

As stated, a combination of somewhat prescriptive safety management activities, generic goals, and process-based guidance must be captured in future standards for them to be effective and to allow a wide range of technologies to be certified. More specifically, it must be recognised that the prescriptive process-based standards are primarily a hindrance for the development and assurance of software, particularly for new and emerging technologies. It is this aspect of safety engineering that needs to be and that will gain the most benefit from a goal-based approach. The Safety Management approach should remain fairly prescriptive, structured and consistent in future safety standards. In fact it is already fairly consistent across existing safety standards from different countries and industries. The objectives and goals of safety management are investigated in more detail in a subsequent section but before this is done, we depict the generic top-level goals that would be applicable to most development projects and that should be reflected in future standards.

5 Generic Goal Structures

Although several standards have adopted goal-based approaches to safety assurance, there are differences in the way the safety argument is constructed and justified. The Goal Structuring Notation (GSN) is emerging as one of the preferred methods for constructing a goal-based argument, and is defined in The Yellow Book (2007).

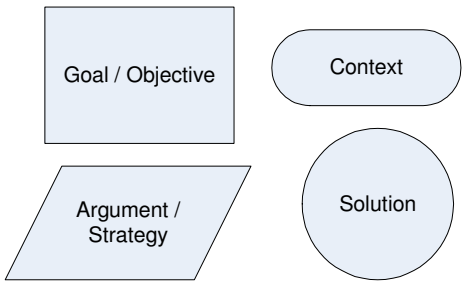

Figure 3: Elements of Goal Structured Notation

The GSN is a graphical notation that explicitly represents the individual elements of a safety argument (requirements, claims, evidence and context) and, perhaps more significantly, the relationships that exist between these elements. That is the GSN depicts how individual requirements are supported by specific claims, how claims are supported by evidence and the assumed context that is defined for the argument. The principal symbols of the notation are shown in Figure 3.

Figure 4 provides an example of a goal structure of safety arguments, which is generally applicable to most applications.

6 Generic Safety Management Goals

As detailed above, the safety management approach should remain prescriptive and consistent amongst future safety standards. This section will expand goal G7 of Figure 4 to define the safety management goals that would enable the other goals to be achieved by ensuring that safety activities are planned, monitored against the plan, and effectively executed.

Practical experience in safety-related systems and research of existing safety standards (e.g. Def Stan 00-56 (2007), The Yellow Book (2007), IEC 61508 (1995), MIL-STD-882C (1996), and Def(Aust) 5679 (1998)) have identified the following key requirements for the development of safety systems.

1. It is essential to have a systematic approach to safety that incorporates techniques which are valid for hardware, operators and software.
2. System design must be inherently safe; issues raised during hazard analyses must be allowed to impact system design if necessary.
3. The use of integrity levels allows the application of techniques and measures which is appropriate to the criticality of a component. A practicable and sound approach is needed for the assessment of integrity levels for system components.
4. A well-defined set of appropriate techniques and measures must be applied to deliver assurance of safety.

It can be seen that these key requirements are reflected in the main safety argument S1 of Figure 4, and are based on:

1. Safety requirements are complete and correct (G2)
2. Safety requirements are satisfied (G3)
3. Appropriate standards applied (G4)

From the surveyed standards, the generic Safety Management goals indentified are:

1. Define Safety Scope: Describe the safety policy, collect information about the system and environment in which it will operate, establish the boundaries of the system and define the scope of the hazard analyses.
2. Define Safety Acceptability / Tolerability Criteria: This must be done in cooperation with the customer. It should be noted that different countries and different industries require the risk scale to be adaptable to suit the particular system

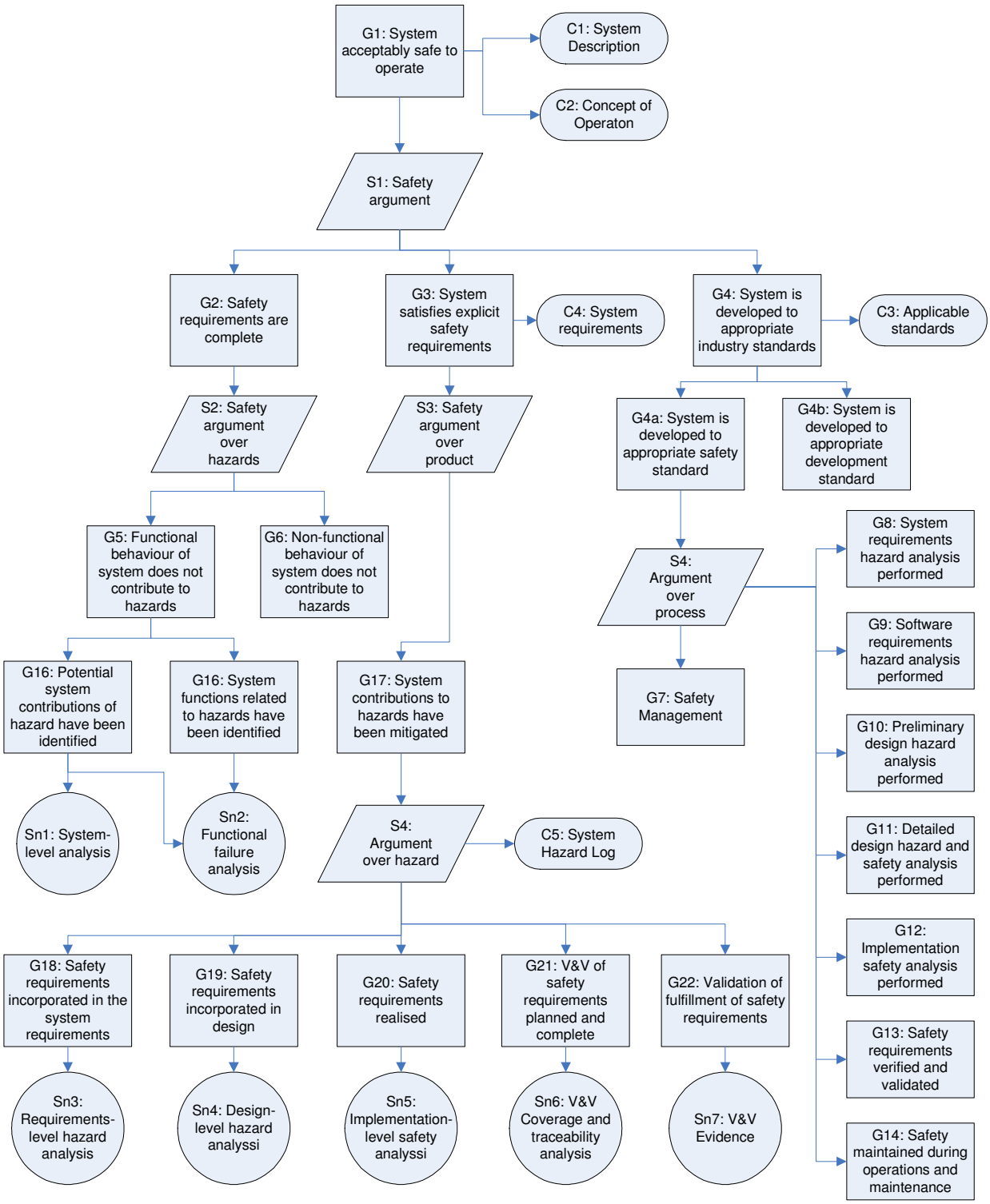

Figure 4: Example Generic Goal-Structured Safety Argument

implementation depending on the operational profile of the relevant system.

3. Define Safety Organisation: Establish and maintain a safety organisation structure for the project, including specifying roles and duties of personnel and groups, providing reporting channels, and ensuring adequate levels of managerial and technical skills and independence.
4. Define Interface to Other Disciplines: Define the interactions and data/information flow to and from other safety disciplines and other system engineering disciplines to ensure they effectively work together and do not duplicate work.
5. Define System Safety Management Plan: Describe the activities for achieving functional safety, plan the safety analyses and assessments,

and describe the means to develop and maintain the Safety Case.

6. Define Hazard Tracking System: Define a single closed-loop hazard tracking system to document hazards from identification to closure, detailing the risk assessment, risk reduction and verification evidence.
7. Establish Safety Management Group: Set up a system safety management group (also referred to as system safety working group and safety committee by the surveyed standards) to oversee, review and endorse safety management and engineering activities.
8. Define Safety Development Assurance Tasks: Define the process for demonstrating allocated integrity / assurance levels of components.
9. Independent Safety Assessment: Plan for and assign an independent organisation to provide assurance that relevant legislations, standards and policies are complied with.
10. Define Safety Management System: Provide a through life safety management plan to manage and maintain the system Safety Case during maintenance and modifications until decommissioning and disposal of the system.

Future safety standards should prescribe the abovementioned system safety management requirements. The main reason why this can be more prescriptive is because it is not technology specific.

The key benefit of the goal-based approach will however be more evident and obvious for the development assurance of software and, to a lesser extent, hardware which are technology dependent.

7 Safety Development Assurance

The primary objective of development assurance is to provide confidence that the system is free from systematic faults. The second objective is to demonstrate that the safety requirements have been correctly implemented. Development assurance is required for the development of software, hardware and configuration data, like application data for a generic product.

Functional safety requirements (generally) require the performance of certain functions to provide a level of hazard mitigation and risk reduction. The safety integrity (also referred to as Development Assurance Level) requirements define these performance requirements, which are directly proportional to the level of risk reduction required and claimed. The higher the level of risk reduction, the higher the level of integrity and confidence required that the component is functioning correctly.

Integrity requirements define the reliability and robustness required for the given safety requirements and can also be used to define the availability of the system to perform its functions.

Standards that use Safety Integrity Levels (e.g. IEC 61508 (1995)), or their equivalent concepts (Development Assurance Levels in SAE ARP 4761 (1996) or Safety Assurance Levels in DEF (Aust) 5679), (1998) explicitly or implicitly define good practice for each of the levels and therefore implicitly link engineering methods and tools with risk and quantitative or pseudo-quantitative requirements. By dictating methods, a strategy for achieving the requisite confidence is imposed, which may work well for some applications but be a hindrance in others as already discussed.

This is why development assurance needs to provide flexibility to allow service providers to select the most appropriate set of techniques and practices for the system under development. We cannot get away from applying a set of techniques and measures to develop software and hardware. But unless the techniques and measures applied are considered to be industry good practice, it will be difficult to justify in the safety argument.

The Yellow Book provides the service provider some flexibility when it comes to development assurance by providing a list of prescriptive process standards (e.g. EN50128 (2001), IEC 61508 (1995)) that may be applied. What would be even more practical is to allow for the service provider to select, mix and match, techniques and measures from various development standards, or wherever current industry good practice is defined. It is acknowledged that this is easier said than done. For this mixing and matching of techniques and measures to be effective, there needs to be a link between development assurance goals and development processes defined within the standards.

The software assurance parts of development assurance standards, like EN50128 (2001), IEC 61508 (1995), DO-178B (1992), DO-278 (2002), Def(Aust) 5679 (1998), SAE ARP 4761 (1996), need to eliminate prescriptive requirements, particularly those that are technology dependent. These standards need to provide a tailorable safety assurance framework that links goals to a flexible development process. The derivation of the framework must focus on safe design concepts (i.e. goal-based) instead of good design practices (i.e. process-based), as design practices are generally tuned towards reliability and quality instead of safety as identified by McDermid (2001).

In addition, these standards need to provide sufficient guidance for alternative techniques and measures that can select in order to achieve these goals for the required integrity. This means a link needs to be provided between goals and development processes to make it easier for service providers to justify that a selected set of processes meets the development goal.

For example, when considering software safety development assurance, the good-practice techniques and measures mandated and/or suggested in the surveyed standards can be categorised into four key objectives or goals:

1. Providing a good design basis for development, customized for safety; expressed as a design and coding standard including selection of a suitable programming language or a safe subset of the programming language.
2. Ensuring that safety requirements are correct and complete; by the application of structured hazard and risk analyses.
3. Ensuring that safety requirements are adequately addressed in the design, and that the code implements only the allocated and derived

requirements; by the provision of traceability and coverage.
4. Providing evidence that each software component meets its allocated safety requirements; by the provision of design and coding verification & validation.

The key generic goals for the development of hardware would be very similar and cover:

1. Quality and Reliability Assurance of Components.
2. Completeness of safety requirements.
3. Requirements traceability and coverage.
4. Design and manufacturing verification and validation.

It is believed that defining generic sets of development goals, particularly for software, as detailed above, is what standards bodies need to focus on in order for the future safety standards to be practical and effective. Generic sets of development goals will most likely need to be defined and fine-tuned for different industries and different types of application to make it easier for the service provider to determine what evidence is required and easier to convince the acceptance authority. This will by no means be an easy activity as much effort and expertise is required to get this right.

8 Show me the Evidence!

The main problem and the question always asked with the goal-based approach, as mentioned already, is "What evidence is needed and how much evidence is enough?" Unfortunately there is no definitive answer to this question. Much effort is required by the service provider to define what evidence will be provided and then convince the acceptance authority. The reason that there is no definitive answer is intrinsic to the goal-based approach in that the evidence required is application specific and specific to the selected method of development. What is clearer is that the amount of evidence required significantly increases as the level of integrity required for (or associated with) the product increases.

Having generic sets of development goals defined, as detailed above, will help by providing a more structured breakdown of the type of evidence required. The service provider needs to break each goal down into manageable sub-goals which in turn make it easier to identify what evidence would support an argument to justify each sub-goal.

Def Stan 00-56 (2007) discusses the need for three types of evidence, and requires that a combination of these need to be provided to justify the overall safety argument; these are: process-based, product-based, and counter evidence based on vulnerability studies. It should be noted that these actually reflect the three approaches described in Figure 1, and are also evident in the generic goal structure shown in Figure 4.

Process-based evidence needs to provide confidence that industry "good" practice was applied for system development and safety management. Generally, product-based evidence is considered to be an output or result of following a particular process. Subsequently having the development processes identified should guide the service provider in identifying the type of product-based evidence that is required for the system under development.

It is important to understand the purpose of the evidence, and what it will be used for. The evidence will be to support arguments about the behaviour of a system to gain confidence that the system is safe. The independent safety assessor will assess each piece of evidence subjectively against each argument by considering:

1. Relevance
2. Sufficiency
3. Argument coverage
4. Validity
5. Independence

As already mentioned, the intrinsic subjectivity of the goal-based approach is the main drawback with this approach. This is why well-defined sets of generic development goals and a consistent safety management approach is so important for reducing some of the subjectivity.

Evidence needs to be placed under configuration management and associated with the system configuration that it allies to. Quality attributes that are associated with most engineering artefacts are likewise applicable to evidence. It must be possible to demonstrate the following properties for each piece of evidence.

1. Existence
2. Precision
3. Completeness
4. Correctness

These will be assessed objectively by the safety assessor.

9 Impact on Existing Safety Standards

So what does this mean for the current popular safety assurance standards (i.e. CENELEC standards EN50126, EN50128 and EN50129, DO-178B, IEC 61508, and The Yellow Book (2007))?

The suggested approach for future safety standards does not necessarily mean that this would be the end of existing standards. In fact, most standards would not require significant change, as large portions are not technology specific and define a relatively generic safety lifecycle and acceptance framework, along the lines of the generic safety argument in Figure 4.

One important change would be the decoupling between these standards, e.g. EN50129 should not prescribe the use of EN50126 (1999).

The Safety Cases approach needs to become goal-based which require the evidence supported safety arguments to be against the behaviour of the system instead of focusing on compliance against the the application of specific development techniques.

The biggest impact would be for the software assurance approach (e.g. EN50128 (2001), IEC 61508 Part 3, DO-178B), which must focus on safe design concepts, covering:

1. Design and coding standard.
2. Application of structured hazard and risk analyses.
3. Safety requirements traceability and coverage.
4. Design and coding verification & validation.

Out of the surveyed standards, the Yellow Book (2007) is the only standard that broadly complies to the concepts discussed in this paper, and hence would require the least change.

1. It is already goal-based and includes the goal structured notation.
2. It will need to allow for flexibility for the selection of development processes.
3. It must define generic sets of development goals, instead of listing prescriptive standards that should be applied.

Some acceptance authorities (e.g. RailCorp, TIDC) are already requiring service providers to provide more evidence to support the assurance argument and not just show compliance to standards and principles. Even without the use of goal-based standards, there will be much more effort required by the acceptance authorities in the future to justify the safety of a design and its implementation. However the goal-based approach will allow service providers to develop the system using techniques that best suit their needs.

10 Conclusions

It should be clear at this stage that prescriptive standards hampers the continual move forward in technology, while the goal-based approach leaves us without suitable advice or agreement on achieving assurance.

A goal-based approach, along the lines of that used in The Yellow Book (2007), has obvious benefits as it imposes fewer constraints on the implementation, both in terms of processes and in technical solutions. The goal-based approach is useful from a safety assurance perspective, as the questions focus on safety-related outcomes (e.g. “what evidence do you have to show that display updates occur within x seconds?).

In a goal-based approach, it is not sufficient to demonstrate compliance to a generic safety process (such as IEC 61508 (1995)). Convincing arguments have to be constructed that relate to the behaviour of the specific product and its safety properties and this can be difficult for service providers to adopt. There is a need to shift from documenting how hard people have tried to develop a system, to providing evidence and arguments about the behaviour of that system.

However, it has to be recognised that such an approach represents a significant shift from:

1. a process compliance approach to a product orientated, safety property approach
2. a tick-box mentality to argument-based mind-set

Safety program management should remain relatively prescriptive. Whereas the future of safety assurance standards needs to be goal-based as prescriptive standards cannot keep up with fast changing technology. For a goal-based approach to be effective and efficient:

1. The goals need to not be technologically specific and focus on safe design concepts.
2. There needs to be a well-defined (somewhat prescriptive) and structured process for safety management, as detailed in Figure 4.
3. Development assurance processes, particularly for software, need to be tailorable and flexible, with a clear link to goals.
4. A rich collection of generic sets of development goals needs to be defined and captured in standards.
5. Guidance needs to be provided for defining the goals and indentifying (and gaining agreement with the acceptance authority) on the type and amount of evidence required.

This shift towards goal-based assurance and arguments will by no means be easy and it will most likely take some time to get things right. A quite a mature industry with lots of experts is required, with the UK leading the way, particularly to develop the generic sets of goals for each industry.

The main challenge with the goal-based approach will be for the service provider and acceptance authority to agree on the goals and required evidence. It is also not clear if the goal-based approach would actually make it easier or more difficult for cross standard acceptance and certification, because of the more subjective nature of the goal-based approach. This requires further research and analysis.

11 References

CENELEC EN 50126 (1999): Railway applications - The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS).

CENELEC EN 50128 (2001): Railway applications - Communications, signalling and processing systems - Software for railway control and protection systems.

CENELEC EN 50129 (2003): Railway applications - Communication, signalling and processing systems - Safety related electronic systems for signalling.

MIL-STD-882C (1996): System Safety Program Requirements. United States of America Department of Defense.

Def Stan 00-56 (2007): Safety Management of Defence Systems. United Kingdom Ministry of Defence.

Def (Aust) 5679 (1998): The Procurement Of Computer-based Safety Critical Systems. Defence Science Technology Organisation (DSTO).

RTCA DO-178B (1992): Software Considerations in Airborne Systems and Equipment Certification. Radio Technical Commission for Aeronautics (RTCA).

IEC 61508 (1995): Functional Safety: Safety Related Systems. International Electro-technical Commission (IEC).

RTCA DO-278 (2002): Guidelines for Communications, Navigation, Surveillance, and Air Traffic Management. Radio Technical Commission for Aeronautics (RTCA).

SAE ARP 4761 (1996): Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipments. Society of Automotive Engineers.

The Yellow Book (2007): Engineering Safety Management, Volumes 1 and 2, Fundamentals and Guidance. Rail Safety and Standards Board. Issue 4.

CAA SW01 (2002): Regulatory Objective for Software in Safety Related Air Traffic Services. Civil Aviation Authority, Safety Regulation Group, Air Traffic

Services Safety Requirement, Document CAP 670, Section SW01.

McDermid, J.A. (2001): Software Safety: Where's the Evidence? *6th Australian Workshop on Industrial Experience with Safety Critical Systems and Software* (SCS'01).