Diametric Software

Commercial in Confidence

"There are two ways of constructing a system design: One way is to make it so simple that there are obviously no deficiencies and the other way is to make it so complicated that there are no obvious deficiencies."

(C.A.R. Hoare, 1980 Turing Award Lecture)

Email: info@diamtetricSoftware.com

Website: www.DiametricSoftware.com

Contents

Executive overview

Today's technically advanced world is delivering more complicated and intricate systems, whether configurations of equipment & machinery or software-based control systems.

The UK's latest aircraft carrier - HMS Queen Elizabeth is a perfect example of this trend. It has more than 70 complex systems and over 1000 equipment types, all with extensive design configurations, detailed operational analysis and complex safety cases. Safety Management is becoming far more complex and challenging and this trend will continue for the foreseeable future.

Today, the best way to demonstrate that a system is safe is through the compilation of a safety case. It provides a clear summary of the operating processes and outcomes required by the competent regulatory or operating body and allows them to make an informed decision on safety.

Safety cases often amount to hundreds of thousands of pages on design & analysis and are compiled with input from many organisations and a myriad of information tools and sources.

The result is often an ad-hoc mixture of text, diagrams and spreadsheets that are difficult and time-consuming to produce and maintain, do not lend themselves to frequent change and are prone to errors. Relevant information is hard to find and the lack of explicit links to data means that arguments must be duplicated in many places.

Operating in this highly complex and fluid environment requires clarity of presentation and effective analysis. Current EA based MBSE are not conducive to effective safety case management. There is a clear need to improve and modernise safety case production by a software-based safety management system that provides a single point of truth.

This first white paper looks at the challenges of safety case production, particularly in the Systems Engineering environment.

The second white paper goes on the show how the Diametric Safety Case Manager address all these issues and provides a bridge into the next generation of safety cases.

The third White paper looks at future development of the software including how safety cases can be linked to ongoing operations and incident/accident reporting.

The safety case

Introduction

A number of major incidents over that last 60 years including the Windscale Nuclear installation, the Piper Alpha Oil rig and two space shuttle disasters, have resulted in the adoption of more stringent safety standards and more importantly, the requirement to demonstrate to a regulatory body that the system is safe.

Initially this constituted a demonstration of adherence to the regulations and procedure supplied by the governing organisation or regulatory body. More recently, with the introduction of safety cases, the onus has shifted towards a more holistic requirement to argue the case that the system is safe across the whole scope of operation.

A clear summary of the operating processes and outcomes is required for oversight by the competent regulatory or operating body, with arguments to demonstrate safety and links to supporting evidence, in order for an assessor to make an informed decision on whether they are safe.

The Safety Case Concept

The safety case must provide an external reader with the logical reasoning as to why the system is safe to operate or, a change to the operating procedure can be considered safe. Documentation should be prepared in sufficient detail so that anyone reading it will be able to see not only what decisions were reached but what the justification was for classifying risks as tolerable and acceptable.

A safety case can cover anything from a simple circuit to a software configuration or a set of operating procedures and it should:

- Communicate a clear argument to another party, from the available evidence, that the system is safe
- Show the context within which the system operates - it is impossible to argue that something is safe without such context
- Convince someone that the system is as safe as is reasonably possible - absolute safety is an unattainable goal

Meeting regulatory requirements

Regulators are increasingly requiring safety cases as a means to demonstrate that the required, tolerable levels of safety have been achieved and legislative & regulatory requirements met. Safety cases provide a clear structure for organisation and are often used to provide the foundation for a future safety management regime.

Constructing a safety case

There are many definitions of what should constitute an effective safety case, but all agree that at the core is a clear and concise argument that demonstrates that the system or equipment is acceptably safe. An effective safety case report should contain the following elements:

Aim:	What is the safety case detailing ie: that the subject of the safety case?
Purpose:	Why is the safety case being produced and for whom?
Scope:	What is being covered and just as important, what is not!
System Description:	A description of the system and changes so as to promote understanding of the safety case itself
Justification:	The regulatory requirement and the reason for changes
Argument:	A well-reasoned and clear argument showing how the aim is achieved
Evidence:	The evidence that supports the claims made in the safety argument
Caveats:	Any assumptions made or limitations and restrictions that apply to the system
Conclusions:	A simple statement to show that the aim has been met

Bringing it all together

The arguments and evidence required to provide an effective and detailed safety case can often be huge. Safety Engineers constructing a safety case are often left with the unenviable task of attempting to present a safety argument that encompasses tens of thousands evidential items.

Safety cases also rely on a vast number of sources to provide the context and evidence required to clearly understand how it has been presented. Giving access to all of these sources requires careful handling if the flow and relationship structure is not to be compromised. A safety case is often the work of a number of teams and consistent cross-referencing without electronic support is hard to maintain.

Presenting the argument on paper

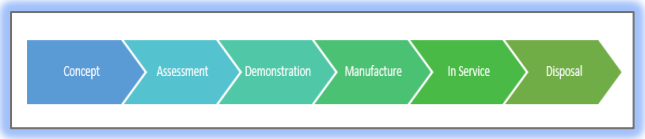
Although it is possible to communicate safety arguments clearly within a paper-based safety case the text must be precise and judiciously controlled, if ambiguity and misunderstanding are to be avoided. Paper based safety cases can amount to hundreds of thousands of pages from a myriad of sources.

A single point of Truth

Paper-based safety cases create a tension between referring to arguments and evidence that is laid out elsewhere, which forces the reviewer to keep flipping between documents, and repeating arguments and evidence every time they are relevant, which increases the size of the safety case and creates the potential for inconsistency during updates.

The safety case should present a single point of truth for each argument and item of evidence. Paper safety cases make this impossible.

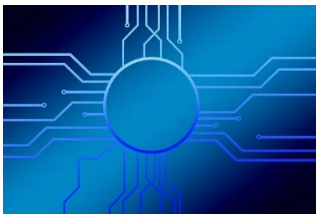
System lifecycles



Previously, safety cases tended to address the system in the design or concept stage but must now be kept up to date throughout the complete lifecycle. Recent practise has shown that the most effective safety cases are those that have been developed incrementally and in parallel with the system or organisation they describe. This will mean that the safety case can no longer sit in a drawer and gather dust; it must be constantly reviewed and managed continuously. Assessing

required changes and implementing them is now a necessary safety function and without software support, this can be extremely difficult and time consuming.

Clarity and traceability



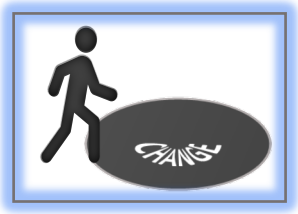
With many safety cases there is a problem with the clarity of the safety argument and the links to other content such as justification, context, assumptions and evidence. The individual threads of a safety argument are often buried in many levels of documentation and sections of the detailed text and thus are not easily traceable or clear to the reviewer.

Reuse of the safety case

The structure and tenets of safety case arguments provide the opportunity to reuse similar parts of an argument particularly across a complex system or organisation where such arguments occur again and again. The advantages of doing this are that time is saved in producing safety case documentation by providing best practise and a template for production. However, care must be taken to avoid misunderstandings or misinterpretations of the context and rationale supporting the claim

Managing change

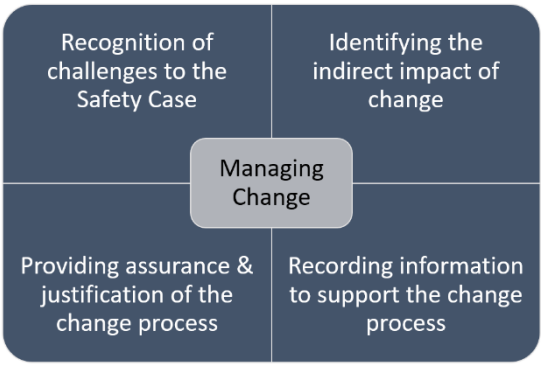
As we have stated earlier, good safety cases follow the lifecycle of a system and will be subject to continuous change. Changes may appear insignificant at first review but on detailed analysis may demonstrate a significant impact on the argument presented. The ability of the safety case manager to identify relationships and references to other aspects of the safety case is a major factor in its effective utilization.



Underlying context, justifications and assumptions also need to be assessed to understand what has changed and the impact of those changes on the safety case. The identification of the impact of change to a Safety Case will then lead to the need for clarity on subordinate or related (directly or indirectly) dependencies. Incomplete or insufficient information will undermine the safety case completely.

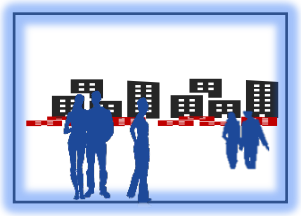
When a safety case audit is conducted, and/or an incident occurs that challenges the validity of a safety case, a review will be required. An assessment must be made as to the soundness of the challenge and the changes required to address the shortfall.

This response will require extensive analysis followed by the implementation of changes to restore validity. Furthermore, these changes must be *readily* available and *clearly* highlighted to a reviewer so that an effective assessment can be made as to the effectiveness of the changes. If this is not the case there is a real danger that the safety case will be rejected, whether it is valid or not.



Goal structuring notation

One way of providing better clarity and understanding of complex safety cases is to use Goal Structuring Notation (GSN). It is a graphical argumentation notation that explicitly represents the individual elements of any safety argument and the relationships that exist between these elements (i.e. how individual requirements are supported by specific claims, how claims are supported by evidence and the assumed context that is defined for the argument).



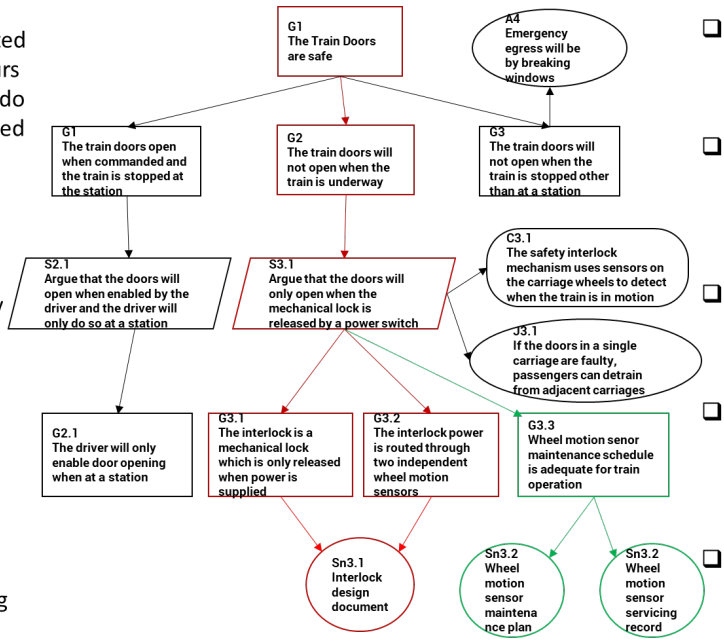
Use of the GSN provides a clear high-level model of the safety case structure that is easy to follow and understand as well as providing a gateway to more detail information. When a safety case is challenged or modified the relevant areas affected can be quickly identified and analysed for deeper or hidden interdependencies that may be affected. This makes the configuration of the safety case much easier.

Using GSN

Building the High Level Argument (HLA) of a safety case can be quickly achieved with GSN as it makes clear the strategy used, the goals (claims) that are being made, the context and justification behind these claims and the supporting evidence (solutions). In a complex system there will be many Goal Structures to build and assess for change and although the GSN helps to quickly understand the goals, contexts, justifications, assumptions and evidence, the difficulties in analysing interactions and links can still present a significant challenge. Moreover, it doesn't stop you writing bad arguments, it is merely a communication tool and not an end to itself.

An example GSN argument for a fictional automatic door is shown below. The GSN allows the selection of high level goals to focus detail in a particular area of the safety case. In the example one aspect of interest – *G1: The Train Doors are Safe* – has been selected for further examination. This could be the result of new equipment being fitted to the configuration or an incident where the mechanism failed and a resultant challenge to the safety case made.

- ❑ A Challenge is instigated when an incident occurs where the train doors do not function as expected
- ❑ What happened?
- ❑ The trains doors were unlocked while the training was underway
- ❑ Identify the key area (Red)
- ❑ What is the strategy
- ❑ Identify subordinate Goals
- ❑ Identify the supporting evidence



- ❑ A failure in the wheel motion sensor was identified as the problem
- ❑ Maintenance and serviceability were reviewed in order to resolve the problem
- ❑ New procedures were put in place to overcome the failing
- ❑ Maintenance plans and serviceability performance documented and added as evidence
- ❑ Safety case updated with new goals, etc clearly identified (Green)

Evidence Assurance

Today, software intensive systems are becoming more complex, under constant change and difficult to analyse. There is a tendency to assess safety and indeed other system properties through observation rather than direct analysis. Observation of incidents or discovering vulnerabilities when the system is already in operation cannot provide the high levels of assurance needed.

Empirical analysis of systems, particularly software intensive systems, does provide evidence that a system meets its desired safety properties, but when used alone, it provides an unsatisfactory and often unclear link to the safety argument. Unstructured hives of evidence will not provide the necessary assurance for a safety case and must be carefully selected using well-structured arguments that show how the evidence relates to and supports a claim. This can then be clearly depicted using GSN.

Types of Evidence

As the initial system design may be highly abstract and incomplete, as will the safety case; at this point, it will be a framework that gives a preliminary indication of the kinds of evidence that are needed to support the safety case. The Through Life-cycle Management (TLM) process generates artefacts that will serve as evidence — requirements documents, design rationale, test plans, test results, and results of software code reviews and these should be linked to claims made in the safety case argument as they are developed.

Analysis Tools & Techniques

There are a number of tools in Safety Management that contribute to the production of evidence to support a safety case argument. Those contained in the Diametric Safety Manager are listed below:

Hazard and Operability Analysis (HAZOP)	Hazard and Operability Analysis (HAZOP) is a technique for identifying and analysing hazards and operational concerns of a system.
Failure Mode Effects Analysis (FMEA)	An FMEA involves reviewing as components, assemblies, and subsystems to identify their failure modes, their causes and the effects on overall system safety.
Bowtie Analysis	Bow-tie analysis diagrammatically represent hazardous events to show the connections between hazards/threats and their consequences.
Risk Matrix	A risk matrix is used during risk assessment to define the level of risk by considering the category of probability or likelihood against the category of consequence severity.
Fishbone Analysis	Fishbone analysis is tool for thinking through all possible causes of a problem.

Model Based System Engineering

Currently, Model Based System Engineering (MBSE) tools do not have the capability to do effective safety assurance. System engineering models are used to provide evidence for the safety case, but they do not, on their own, provide a detailed and coherent safety argument.

In theory it should be possible to extend an MBSE tool to store safety arguments. In practice we have found this is not feasible:

- Reporting in current MBSE tools like Sparx Enterprise Architect is inflexible. Customised reports require hand-coded Visual Basic (VB) macros to generate, with minor adjustments often requiring further VB changes. This often leads to software bugs and lengthy delays which are not conducive in a safety management function.
- The Enterprise Architect UML meta-model is not aligned for safety case information. GSN consistency rules cannot be implemented and the user has to trawl through a large number of non-safety related fields and select other fields to enter safety data which are not immediately obvious.

There is clearly a need for an improved safety function within current MBSE tools or interaction with stand-alone safety management systems.

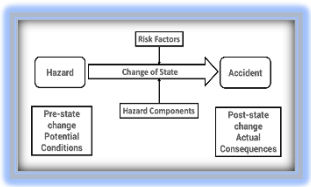
The Formal Safety Assessment

A huge amount of analysis will be required for a system safety case and a Formal Safety Assessment (FSA) will be required. Elements of an FSA are detailed below and clearly demonstrate the complexity of building and updating a safety case:

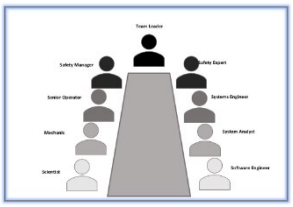
- Identification of safety risk acceptance criteria - who does it, what level of risk is acceptable, Rules for prioritising the risks
- Identification of the boundaries of the analysis - System, mission phases, lifecycle etc, Hazard, severity levels etc, Qualitative versus Quantitative Analytical Techniques
- Identification of the accidents - top level events
- Identifying Hazards and Targets
- Linking Hazards to the Accident - the Sequence
- Assessing risks - causal factors, probability, severity etc
- Identifying controls – performance standards for procedural controls, costs of controls
- Documenting in a hazard log - PHA form can be used
- Reviewing and evaluating overall residual risk
- Preparing the capstone high level argument – concise, from the facts, identifying assumptions and limitation

Identification of hazards

Hazards are essentially the cause, or the precursor, for accidents. The identification and subsequent management of hazards is a key safety aspect for any system and



there are a large number of techniques available but with the exception of HAZOP, there is little formal guidance. The primary purpose of a HAZOP is to identify deviations from design intent that can lead to the occurrence of an undesired event or hazard. It will require an experienced Team Leader and the use of design representation aids such as functional block diagrams, reliability block diagrams, context. It is a time-consuming process, particularly if many people are involved in the braining storming sessions and it is essential to record and refer to the output. Once hazards have been identified, they must be logged and tracked throughout the life of a safety case. A challenge to a safety case could involve a review of existing hazards and a further HAZOP may be required. The whole process and output is extensive



Summary

Safety cases are still evolving and developing in the industrialised world, but there is no doubt that their complexity is continually growing. So too is the challenge to manage them and the summary below captures many of the real challenges facing system engineers today.

Observations	Implications
Identifying the key elements and their interdependencies is a key factor in a Safety Case.	• *Can be time-consuming* • *Can lead to misunderstanding and false analysis*
The detail and evidence in an effective safety case is often extensive.	• *Safety Engineers need to present and link tens of thousands evidential items* • *Consistency is hard to maintain*
Safety cases often have a vast number of sources in order to provide the necessary context and evidence.	• *Very careful handling is required if the flow and relationship structure is not to be compromised* • *Significant cross-referencing the work of many contributing individuals is difficult to implement and maintain*
Current MBSE tools are not effective at managing safety cases	• *EA UML not consistent with GSN*

	• *Report generation requires lengthy code changes* • *EA fields not conducive to safety data*
Paper based safety cases can amount to hundreds of thousands of pages from a myriad of sources.	• *Text must precise and judiciously controlled if ambiguity and misunderstanding are to be avoided* • *To provide clarity and continuity, a considerable amount of repetition can occur reproducing text from other sections* • *Referring to and sorting out a large number of references is time consuming and can break the thread of the argument in the reviewer's head*
Changes must be meticulously recorded in a safety case.	• *If focus and true meaning are not to be lost, changes must be properly tracked and annotated in other relevant parts of the document*
A safety case must be maintained throughout the System's lifecycle.	• *Information must now be relevant throughout the complete lifecycle of the system* • *Safety cases should be developed incrementally and in parallel with the system's development* • *Constantly review for relevance and change will require regular reviews and updates*
Safety Cases require good clarity of the safety argument and links to evidence.	• *Individual threads are often buried in many levels of the documentation and are often not easily traceable or clear to the reviewer*
Reuse of an argument from a previous section of safety case can save a considerable amount of time and effort.	• *Care must be taken to avoid misunderstandings or misinterpretations of the context and rationale supporting the claim*
Changes to a safety case may appear insignificant at first review but may have a significant impact on the argument.	• *A detailed analysis should be conducted on any change, however minor* • *Key relationships and references to other aspects of the safety case must be identified*

	• *Underlying context, justifications and assumptions need to be assessed for impact*
When a safety case audit is conducted, and/or an incident occurs that challenges the validity of a safety case, a review will be required.	• *An assessment must be made as to the soundness of the challenge and the changes required to address the shortfall* • *This response will require extensive analysis of the areas affected by the challenge* • *Changes must be implementation to restore the validity of the safety case* • *Changes must be readily available and clearly highlighted to a reviewer*

From conversations with a number of Safety engineers, it is apparent that many safety cases consist of an ad-hoc mixture of text, diagrams and spreadsheets. These are difficult and time-consuming to produce and maintain, do not lend themselves to frequent change and are prone to errors. Relevant information is hard to find and the lack of explicit links to data means that arguments must be duplicated in many places.

There is a clear requirement to improve and modernise safety case production and operating in this highly complex and fluid environment requires clarity of presentation, effective analysis and a single point of truth. In our second white paper we address how the Diametric Safety Manager achieves this and addresses the many issues faced by system engineers today.

Andrew J King

Paul Johnson

Diametric Software

References

- *Evidence of Assurance: Laying the Foundation for a Credible Security Case – Charles B Weinstock/Howard F Lipson*
- *Electronic Safety Cases: Challenges and Opportunities – Trevor Cockram / Ben Lockwood*

- *A Systematic Approach to Safety Case Management – Dr Tim Kelly*
- *Managing a System Safety Case in an Integrated Environment – Saeed Faraooy*
- *Safety in engineered systems - A discussion at The Royal Academy of Engineering*
- *Systematic Maintenance of Safety Cases to Reduce Risk – Omar Jaradat / Iain Bate*
- *Review of Hazard Identification Techniques – Health and Safety Laboratory*
- *Hazard Analysis Techniques for System Safety – Clifton A Ericson II*