Modular Safety Cases

Facilitating Incremental Upgrade to Military Capability

by Managing the Complexity of Safety Assurance

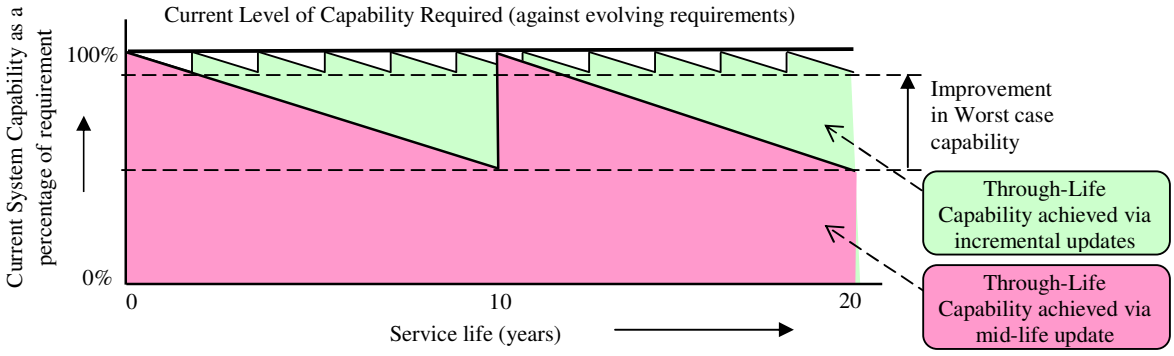
Executive Summary

Maintaining military capability at 'state of the art' levels, through-life, requires a more cost-effective approach to modifying defence systems. A significant proportion of change cost is attributable to recertification of safety aspects of the changed system. The use of modularity provides a mechanism for managing the complexity of Safety Cases and limits the impact when system changes are introduced, reducing the overall through-life costs of safety certification. Although the principles of modular Safety Cases, as described here, could be applied to most defence systems, a specific process has been defined, matured and validated for software-intensive systems. An overview is provided of the Modular Software Safety Case (MSSC) process and issues for consideration when deploying this process are discussed.

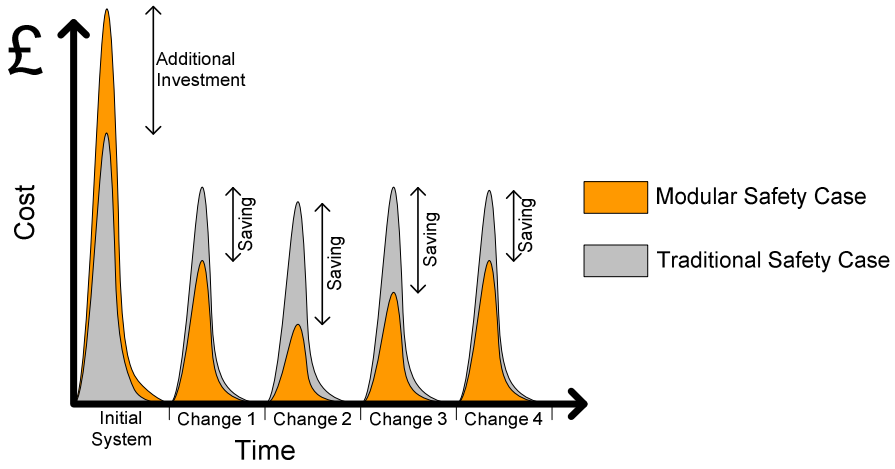
Whilst the process was developed with military applications in mind, the principles and process are also amenable to adoption in a non-military context.

Introduction

The frequency and scope of changes to defence systems are typically constrained by the significant costs involved, yet the military capability provided by a system is perceived to decay over time when it is compared to the evolving 'state of the art' systems which may be available to cooperating or opposing forces. Rather than addressing short-falls against a perceived and constantly evolving military capability requirement, changes are 'parked' until they can be collated into a major upgrade programme, such as an aircraft mid-life update. If costs could be reduced, more frequent, smaller changes could be deployed, more closely 'tracking' the peak operational capability, as illustrated below:



The safety certification for modified systems is a major contributor to the cost of change. A modular approach reduces the cost of re-certification of changed systems, offsetting any 'set-up' costs and leading to an overall through-life cost saving, when compared to traditional Safety Cases. This is shown indicatively below:



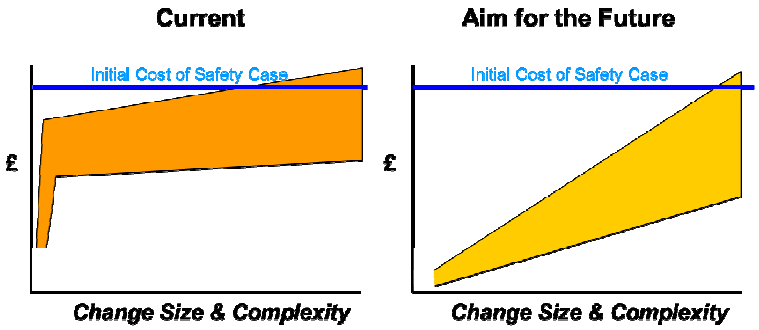
As of the end of 2012, a modular Safety Case process has been trialled on real defence software systems. The software aspects of the process have been matured and validated to a level suitable for immediate deployment for software-intensive defence systems. Modular principles are currently being adopted for system safety assurance and performance qualification on several aircraft programmes. This modular approach has been demonstrated on military avionics, but the principles could equally be applied on other defence as well as civilian systems.

Software Safety Cases

Over the last few decades, systems have become increasingly dependent upon software to deliver defence capability. The size and complexity of the designs that underpin the software on those defence platforms has grown, whilst, at the same time, the need to demonstrate that the system will behave in a safe manner has never been more pressing. DEF STAN 00-56 requires a Safety Case for all defence systems and characterises it as:

*"The **Safety Case** shall consist of a structured argument, supported by a body of evidence, that provides a compelling, comprehensible and valid case that a system is safe for a given application in a given environment."*

The cost of generating a Safety Case for a complex defence system is significant, as is the cost of revising that Safety Case if the system is changed. Due to the complexities of many systems, it is common that, for all but the smallest of changes, the cost of revising the Safety Case for a changed system approaches the original production cost for the Safety Case, bearing no direct correlation with the change size. The left-hand diagram indicates typical distribution of the costs for re-certifying changed systems:



A powerful approach when dealing with complexity, in many hardware as well as software designs, is to break a design down into smaller, more manageable parts, or 'modules'. The current approach to writing Safety Cases focusses on complete 'systems' and does not take best advantage of this design 'modularity'.

The approach described here recommends applying a similar 'modular' approach to the Safety Cases that are used to assure the safety of defence systems and particularly those produced for software-intensive systems. The intention is that, by managing complexity, the cost of revising the Safety Case for a changed system should be more closely related to the size and complexity of the change itself, rather than the size and complexity of the system being changed. This desired cost distribution is indicated on the right-hand diagram above.

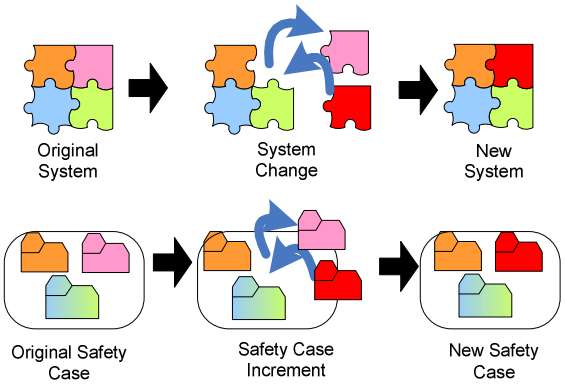
What are Modular Safety Cases?

Modular Safety Cases take advantage of the modularity in system designs, allowing assurance of the safety of a system that has been composed from these design modules.

The principles that are applied to modularising designs can also be applied to modularising safety arguments. Indeed, the approach is most effective when both are considered in parallel. The impact of predicted future system changes should be considered on both the design modules and their associated Safety Case Modules, such that the placement of module boundaries can be optimised for resilience to change.

Modules in the Safety Case are mapped onto single, multiple or even part of design modules which have well-defined safety properties. Defining the safety-related properties of a design module, as well as any dependencies they have on other modules within the system for supporting the delivery of these safety properties, provides a valuable mechanisms for understanding the safety influence of each module, or groupings of modules. The disciplined recording of the interconnections and dependencies in the design domain provides the under-pinning for the safety assurance claims about the design modules. When a system is composed from these design modules, there is then a clearer understanding of the safety dependencies between the design modules and between the Safety Case modules relating to them. The principle is illustrated below where the shapes of the jigsaw pieces are comparable to the defined interfaces between design and Safety Case modules.

The improved understanding of the inter-dependencies in the design domain, with respect to delivery of system safety properties, provides the mechanism to better understand and manage the impact of any change to the system.



Although the principles of modular Safety Cases, as described here, could be applied to most defence systems, a specific process has been defined, matured and validated for software-intensive systems, the Modular Software Safety Case (MSSC) process. The MSSC process, [see Ref 1], requires recording of the safety properties at the interface between selected groups of software modules which are then used in the corresponding Safety Case Modules. An overview of the process is provided in the Appendix of this document.

Goal Structuring Notation, (GSN), [see Ref 2], is a commonly used notation for presenting the structured argument of the Safety Case and has been extended for use with modular Safety Cases, as part of the MSSC process definition. GSN 'patterns' are included with MSSC, to describe commonly-used argument forms that may be re-used when the process is deployed. These patterns have been trialled on real defence systems and reviewed by potential industrial users and Independent Safety Auditors, and hence are considered to represent current 'good practice'.

What are the Benefits of MSSC?

Modularising designs, and Safety Cases, provides a range of benefits. Essentially, closely-related features can be considered together, minimising the necessary interaction between modules. This ensures that effort is focussed on examining and defining these interactions. Applying a more rigorous approach to understanding how these design interactions impact on safety may uncover design issues in itself, but also provides the 'infrastructure' for the following:

- ***Module Replacement*** – if a design module is replaced by another design module which meets the same interface safety properties, the system Safety Case is still valid once any Safety Case Modules associated with the replacement design module are available and accepted
- ***Change Containment*** – when a changed design module does not meet the safety properties required by the system, the impact of that shortfall can be tracked through the system via its impact on the safety properties of interfacing design modules. If the change had been anticipated as part of the modularisation effort, only a small number of (design or Safety Case) modules should be affected. The Safety Case for the changed system will:
    - Only need to revise those Safety Case Modules affected by the change
    - Not need to consider the Safety Case Modules where the interface has not been affected
    - Resulting in significantly less effort (cost) and time to update

- ***Distributed Team-Working*** – authorship of a Safety Case can be partitioned effectively, using the same rigorous interface definitions as the Safety Case Modules. This can be important for either:
    - Internal resource management; or
    - Supporting work-share between prime and sub-contractors
- ***Protection of Intellectual Property*** – the implementation details of the design and Safety Case Module can be hidden from readers, whilst the interface properties that affect safety are still exposed. This may also be necessary for dealing with export controls.

The more rigorous approach to interface definition for safety properties may require additional effort when compared with traditional approaches. This effort will be recovered by the significantly reduced cost of generating Safety Cases for the system when changes are introduced throughout the lifecycle of the defence system.

What is the Maturity of MSSC?

The basic concepts of MSSC have been demonstrated in a representative environment, i.e. Technology Readiness Level TRL7, through the use of industrial case studies. Refinements to the process have been developed and continue to be validated by on-going case studies.

How has MSSC been Validated?

The MSSC process was developed using a rotary-wing aircraft system example, validated by developing a comprehensive modular Software Safety Case for a complex avionic equipment on a fixed wing aircraft project and further matured using rotary-wing aircraft examples. The systems to which MSSC has been applied, to date, have been medium or low assurance systems.

Individuals who perform the role of Independent Safety Auditor both within industry and on behalf of MOD have reviewed both the approach and the comprehensive modular Software Safety Case, accepting the overall approach as suitable. The approach has not yet been submitted for a system that is proposed for Release to Service, however, it is proposed for use on a future rotary-wing aircraft project.

What Tools are Available to Support MSSC?

MSSC does not mandate the use of proprietary tools, however the examples used in the process document are presented using Goal Structuring Notation, (GSN). Tools that support GSN are listed on the GSN website, [see Ref 2].

Who has used MSSC?

MSSC was developed by the Industrial Avionics Working Group, which is an industrial collaboration between: AgustaWestland, BAE Systems, General Dynamics, GE Aviation and Selex Galileo. It has been used by these companies for research projects and is proposed for future aircraft project use.

Deciding to Use MSSC

The benefits of MSSC are maximised if the system or platform integrator broadly applies the approach across major systems that are expected to change throughout the lifetime of the platform. This provides cost-savings to the platform customer, but also, in facilitating more frequent smaller capability upgrades, provides a compelling business opportunity for

industry. Consequently, MSSC is suitable for deployment through 'top-down' contractual mandate by Customers or 'bottom-up' offerings from industry.

Criteria have been identified which help to determine whether a project is 'receptive' to modular Safety Cases. These criteria are based on what benefit can be expected from MSSC and how easily a modular Safety Case can be implemented:

Maximising Benefits:

- System Size and Complexity – MSSC is most beneficial in managing change in large and/or complex systems
- Anticipated Change – MSSC delivers greatest benefit where there are expected to be a number of small to medium size changes implemented throughout the life of the system
- Safety Assurance/Integrity – MSSC is particularly suited to systems designed to support software with mixed safety assurance/integrity requirements

Factors Determining Effectiveness:

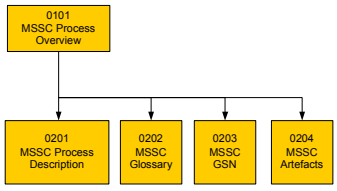
- Design Modularity – MSSC is simpler to introduce for systems that use modular design approaches
- Reusability – MSSC supports the re-use of existing design modules that have an existing safety argument
- Use of COTS/Legacy/3rd Party Software – MSSC can facilitate the use of COTS, Legacy or 3rd party software, provided there is suitable access to design information and evidence

For example, little advantage would be gained from using MSSC for simple, low safety assurance systems designed by a single team. Maximum benefits can be expected where a large, complex system can be modularised with safety in mind, keeping assurance of low integrity and/or low change modules isolated from assurance of modules expected to undergo high change and/or have high integrity requirements.

More guidance is provided on determining the 'receptivity' of a project to MSSC on the Capability Agility Website, [see Ref 1]. A spreadsheet-based tool has also been developed to assist potential users to assess the suitability of their own project and is available at the same website.

Where to find out more

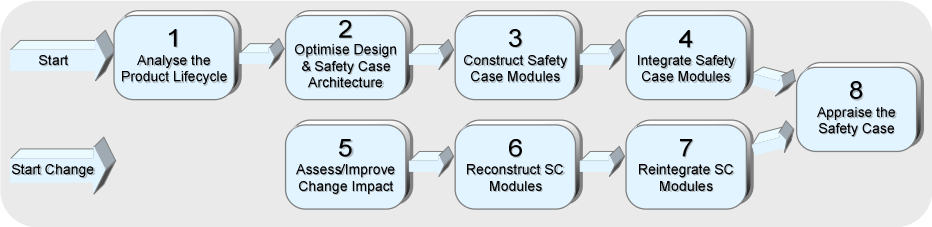
The attached appendix provides a high level overview of the MSSC process. More details and copies of the process documents and guidance, as shown below, can be found at the Capability Agility website, [see Ref 1] or contact david.short@baesystems



References

[1] Capability Agility – www.capability-agility.co.uk
[2] Goal Structuring Notation – www.goalstructuringnotation.info

Appendix – Overview of the Modular Software Safety Case Process



The Modular Software Safety Case (MSSC) presents the development and change of modular Safety Cases in eight steps, five are used in initial development of the Safety Case, and four for incorporating a change. The final step in each is common.

Step 1 : Analyse the Product Lifecycle

The first step in the MSSC Process is to define significant future change scenarios expected during the lifetime of the product which will necessitate the Safety Case being re-visited. These change scenarios provide the driving criteria for choosing both design and Safety Case Module boundaries.

Step 2: Optimise Software Design and Safety Case Architecture

An initial safety case architecture is defined which identifies the safety case modules and their interactions. This is then considered alongside the proposed design architecture and the change scenarios from Step 1 to predict the impact of expected changes. Through review of alternative design or Safety Case module options, and iterative evaluation, a selection is made based on optimised resilience to change. Methodology suggestions are provided in the MSSC process.

Step 3: Construct Safety Case Modules

The Safety Case Modules identified in Step 2 are developed in this step. A hazard mitigation argument is formed and mitigation requirements directed to Block SC Modules. The guaranteed behaviour offered by each block in support of these is captured, along with dependencies on other blocks. A Block Safety Case Module is constructed providing argument and evidence for each Block based on the Guarantees and Dependencies. Additional SC Module types may also be constructed to cover Configuration Data, Software System Wide issues.

Step 4: Integrate Safety Case Modules

The Safety Case Modules are integrated so that claims requiring support in one Safety Case Module are linked to claims providing that support in others. This step of the process results in a fully integrated Safety Case.

Step 5: Assess/Improve Change Impact

When a system change is implemented, the impact on the design modules and associated Safety Case Modules is assessed.

Step 6: Reconstruct Safety Case Modules

This step mirrors Step 3, but is undertaken for the new or changed Safety Case Modules only.

Step 7: Reintegrate Safety Case Modules

This step mirrors Steps 4, but is undertaken for the new or changed Safety Case Modules only. A 'change argument' is also produced addressing the adequacy of the change impact analysis process and the rationale for not re-visiting unaffected Safety Case Modules.

Step 8: Appraise the Safety Case

The purpose of appraising the completed Modular Safety Case in this step is

- Local process improvement and
- to evaluate the Safety Case generated using the MSSC Process to ensure that it is potentially capable of meeting the anticipated safety and cost related objectives throughout the expected lifetime.