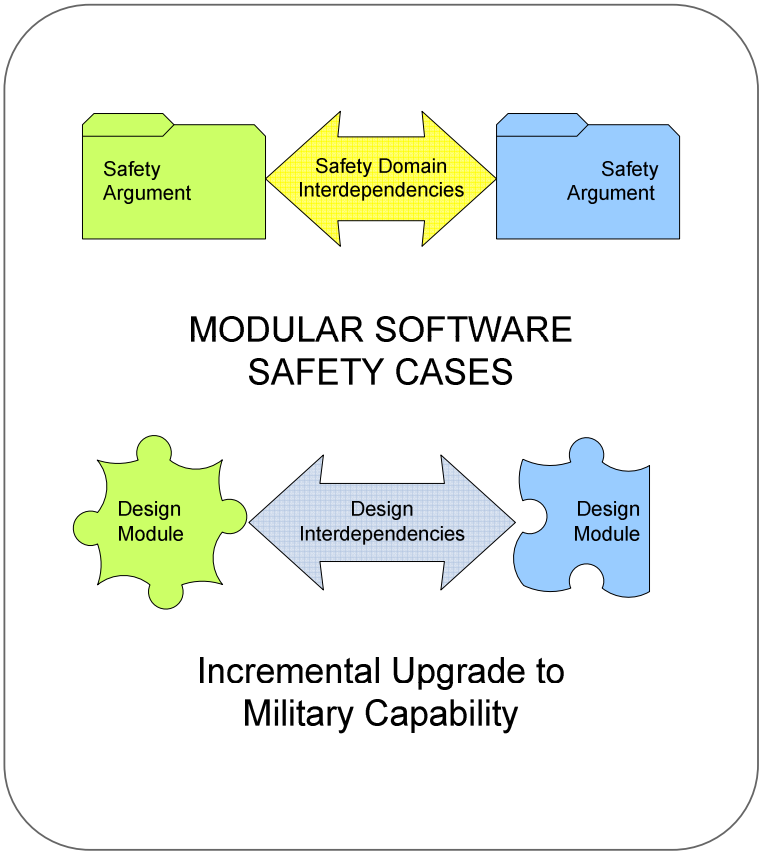
Modular Software Safety Case Process

Artefacts

Date: 19-Nov-2012



Contents

Figures

Tables

1 Introduction

1.1 Purpose

The purpose of this document is to give an overview and explanation of the main artefacts involved in the MSSC process. This document does not describe the MSSC process or the relationship between the artefacts, it merely describes the artefacts in sufficient detail to allow a reader to comprehend the various documents and reports that may be presented as part of a Modular Software Safety Case.

1.2 Document Overview

This document describes the artefacts that are either produced-by or used-by the Steps in the MSSC process.

Section 1 Provides an introduction to the document.

Section 2 Describes the context in which the artefacts are used.

Section 3 Provides a description of each artefact listed alphabetically.

1.3 References

The following documents are referenced from within the text:

[201] MSSC 201 – MSSC Process Overview, Current Issue Applies

[202] MSSC 202 – MSSC Glossary, Current Issue Applies

1.4 Glossary

All the terms used by the MSSC process are defined in the glossary at reference [202].

1.4.1 Abbreviations

All the abbreviations used in this document are defined in the glossary at reference [202].

2 Context

The artefacts are either produced-by or used-by the Steps in the MSSC Process. These artefacts could be documents, entities in a database, a UML diagram or any other suitable medium. It is for the project using the MSSC Process to decide on the medium to use. The MSSC Process is composed of eight steps:

Step 1. Analyse the Product Lifecycle.
Step 2. Optimise Design and Safety Case Architecture.
Step 3. Construct Safety Case Modules.
Step 4. Integrate Safety Case Modules.
Step 5. Assess/Improve Change Impact.
Step 6. Reconstruct Safety Case Modules.
Step 7. Reintegrate Safety Case Modules.
Step 8. Appraise the Safety Case.

The MSSC Process is carried out each time a product needing a safety case is developed and released. A flow chart summarizing the process for the production of the initial Safety Case and how the various artefacts are referenced is in Figure 2-1. Any subsequent formal issue of the Software product will precipitate the need to revisit and where necessary revise the Safety case. A flow chart summarizing the process for the production of any subsequent Safety Cases and how the various artefacts are affected is at Figure 2-2. A description of the MSSC Process is outside the scope of this document and is covered in reference [201]

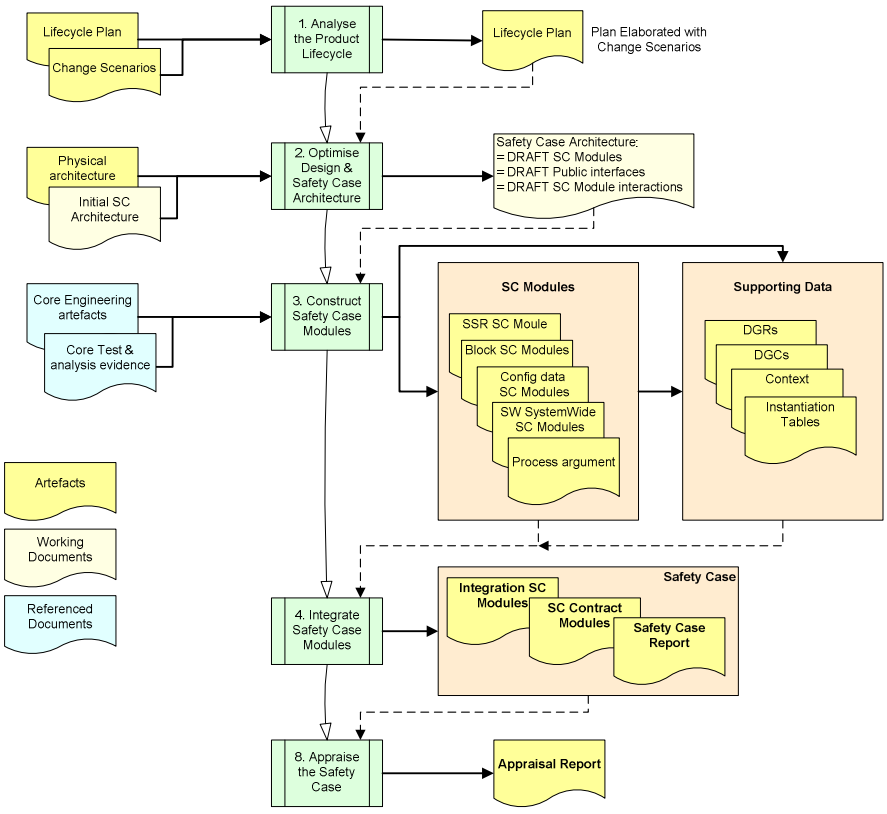


Figure 2-1 MSSC Process Flow Chart – Construction of the Initial Safety Case

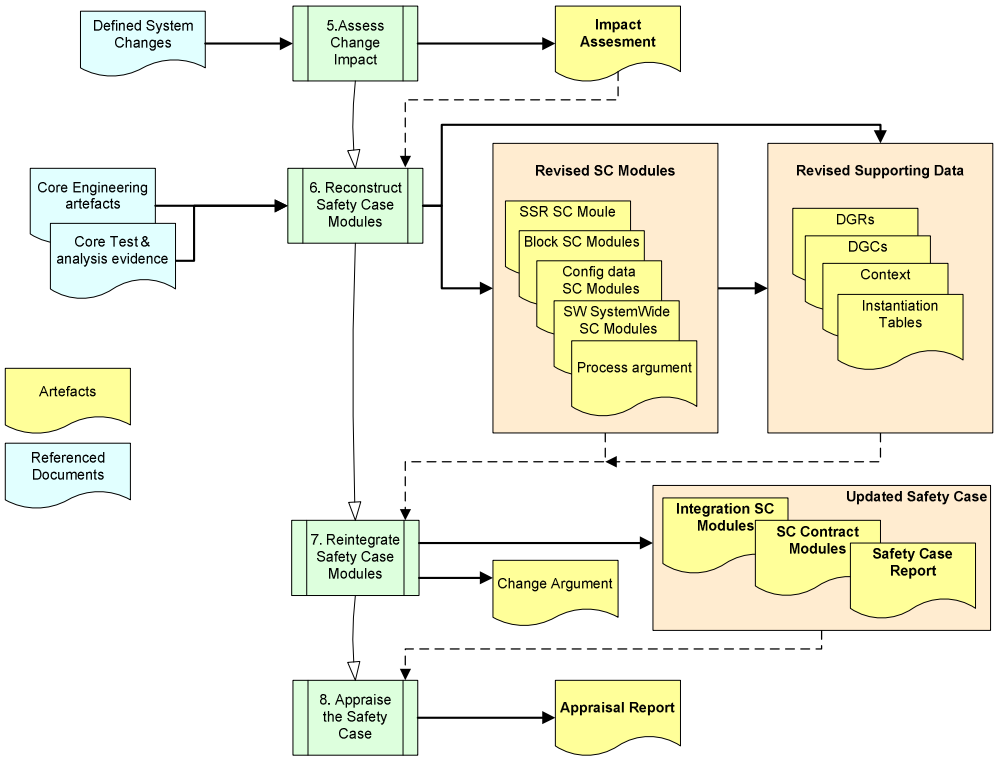


Figure 2-2 MSSC Process Flow Chart – Subsequent Updates of the Initial Safety Case

If the changes are extensive it may be necessary to review and potentially revise artefacts identified in Figure 2-2. In such cases guidance will be provided in the MSSC Process Description document at reference [201].

3 Descriptions

These descriptions of MSSC artefacts should be read in conjunction with the MSSC Process Description [201] which explains how they are used. To facilitate easy cross referencing, the artefacts are listed alphabetically:

3.1 Appraisal Report

This artefact is an appraisal of the contribution of MSSC to the adequacy of the Safety Case and the efficacy of the MSSC process.

The Appraisal Report should state whether the Modular Safety Case adequately addresses the anticipated product lifecycle in terms of the Change Scenarios identified. If one or more Change Scenarios have not been considered in the Modular Safety Case then the report should declare the case as inadequate. As the product develops there may be several issues of the safety case, any divergence between the safety case architecture and the emerging product lifecycle should be identified in the report. The report is an opportunity to assess the complexity of the safety case and state whether or not the arguments contained therein withstand the scrutiny of a third party. Moreover the report should be able to declare how successful the case has been at containing the arguments and constraining them to specific modules. A safety case where the same argument is seen to span several modules should be identified in the report and the safety case potentially declared as inadequate.

The Appraisal Report should also identify the efficacy of the process used to develop the Modular Safety Case, both in terms of the construction of the initial safety case architecture and how it has developed over the lifecycle of the product but also in the application of the MSSC process itself. This may result in recommendations to revise the development practices to better address emerging safety case issues or proposed updates to the project risk register which identify areas where the application of the current Modular Safety Case may be becoming inappropriate. In such cases the report should identify areas where the product design or safety case architecture could be changed to better match the product Lifecycle Plan.

	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8
Produced by								✓
Refined by								
Used by								

Table 3-1 Steps associated with the Appraisal Report Artefact

There would be one Appraisal Report produced for each issue of the Safety Case Report.

3.2 Block Safety Case Module

There will be a Block Safety Case Module for each block identified in the Physical Architecture.

The Block Safety Case Module constitutes some Argument (in these documents this is presented in GSN) and the Evidence that supports it.

The argument has a focal subject which is likely identified by the name of the SC Module.

In the MSSC derivative of modular GSN that this documentation uses a Safety Case module appears as shown as in Figure 3-1.

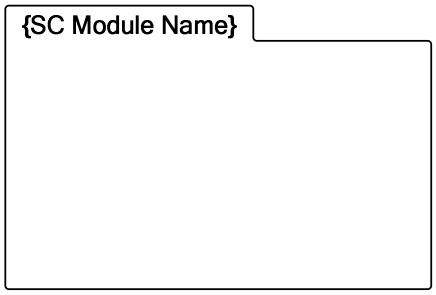


Figure 3-1: Appearance of MSSC GSN part of a SC Module

	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8
Produced by		✓	✓					
Refined by			✓			✓	✓	
Used by				✓	✓	✓	✓	✓

Table 3-2 Steps associated with the Block Safety Case Module Artefact

There will be at least one but probably numerous Block Safety Case Modules in a Safety Case.

3.3 Change Argument

This artefact presents an argument that an updated Safety Case has maintained the integrity of the original Safety Case. It provides the reasoning behind the claim highlighting the Dependency-Guarantee Contracts that have not changed and those that have.

A generic Change Argument would be presented in GSN.

	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8
Produced by							✓	
Refined by								
Used by								✓

Table 3-3 Steps associated with the Change Argument Artefact

There would be a Change Argument artefact produced for each Safety Case increment. Each increment may contain many changes.

3.4 Change Scenario

This artefact describes a potential change scenario for the Software Product. Potential updates and modifications should be captured as a series of Change Scenarios. The analysis of a major Software product may generate dozens of Change Scenario artefacts. These Change Scenario artefacts can then be documented as part of the Lifecycle Plan.

Typically one would categorise Change Scenarios as:

- Known – Arising from planned updates.
- Predicted – Arising from experience of similar systems.
- Unknown – Arising sporadically due to the unique nature of the Software product or the means by which it has been deployed.

	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8
Produced by	✓							
Refined by								
Used by		✓						

Table 3-4 Steps associated with the Change Scenario Artefact

It would be normal to have several Change Scenarios associated with a Safety Case.

3.5 Configuration Data Safety Case Module

In a system that is configured by reference to a set of configuration data it will be necessary to argue that the configuration data is both valid and correct. This is most easily handled as the subject of a Safety Case Module in its own right, which will be the responsibility of the System Integrator to assure.

	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8
Produced by			✓			✓		
Refined by						✓		
Used by			✓	✓	✓	✓	✓	✓

Table 3-5 Steps associated with the Configuration Data Artefact

3.6 SC Contract Modules

These artefacts are used to formalise and argue the satisfaction of public claims in one Safety Case module by claims in another that may not be exactly equivalent. The arguments in a Contract Module justify why the claims providing support satisfy each of the specified claims requiring support.

	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8
Produced by			✓			✓		
Refined by						✓		
Used by			✓	✓	✓	✓	✓	✓

Table 3-6 Steps associated with the Contract Module Artefact

3.7 Dependency-Guarantee Contract

The association between Block Modules where one Block Module's Dependency is satisfied by another's Guarantee is referred to as a Dependency-Guarantee Contract.

Dependency-Guarantee Contracts are often shown on diagrams that use circles to represent the Block Safety Case Modules.

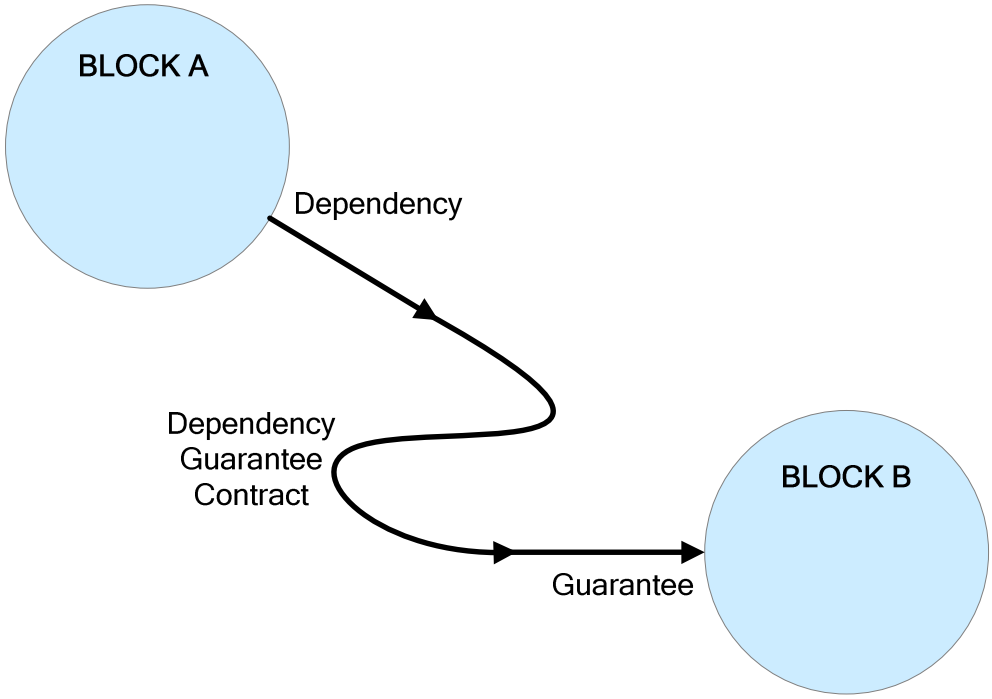


Figure 3-2 Dependency-Guarantee Contract

It may be useful to capture these contracts in a tabular form:

Dependency – Guarantee Contract			**<Block Name>.<DGC Name>**
Consumer Dependency	**Integrator**	✓	**Provider Guarantee**
<Block A Name>.<DGR Name>.<Dependency Number>	**has SC Contract with**		**<Block B Name>.<DGR Name>.Guarantee**
<*Concise Definition* as in the DGR>	is supported by		<*Concise Definition* as in the DGR>
<other *Definitive Context* as in the DGR>	is consistent with		<other *Definitive Context* as in the DGR>
…	is consistent with		…

Table 3-7 Dependency-Guarantee Contract Template

As well as providing information for the instantiation of the argument in a SC Contract, the central column may contain evidence used by instances of the SC Contract. The integrator may record in this column that expert judgement was applied and found no inconsistencies between the Consumer and Provider entries in each row of the table. The approach to

identifying inconsistencies is described under in the MSSC Process Description at reference [201].

If the context includes the data address used by the blocks this may also be confirmed as correctly related to a channel or shared address in the integrators domain.

The form of the DGC table may be optimised and simplified depending upon what information is important to the safety of the particular system and the strength of assurance to be given for context compatibility. For example;

- The full consumer and provider *concise definitions* might not be included in the table, as they duplicate what is captured in the consumer's dependency and the producer's guarantee, which are already referenced.
- The references may be to dependencies and guarantees that are recorded separately rather than in DGRs.

	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8
Produced by			✓			✓		
Refined by						✓		
Used by			✓	✓	✓	✓	✓	✓

Table 3-8 Steps associated with the Dependency-Guarantee Contract Artefact

It would be normal to have one Dependency-Guarantee Contract for each pair of interdependent Block Safety Case Modules.

3.8 Dependency-Guarantee Relationship

This artefact encapsulates a Dependency-Guarantee Relationship for a Block Module. A Block will guarantee to exhibit specific behaviour provided specific dependencies hold. If the dependencies on which the Block relies do not hold then it cannot uphold its guarantees.

For a safety case argument to use the guarantees provided by a Block it will be necessary to confirm that the dependencies the Block relies on have been satisfied. It may be useful to capture these relationships in a tabular form:

Dependency – Guarantee Relationship	**<Block Name>.<DGR Name>**

Guarantee			
Concise Definition	**Definitive Context**	**Incidental Notes**	**Traceability**
<A description of the guaranteed behaviour or property>	<Any from the list of definitive context, below>	<Non-definitive information the author considers informative.>	<References to source material.>

Related Dependencies				
N°.	**Concise Definition**	**Definitive Context**	**Incidental Notes**	**Traceability**
1	<A description of the behaviour or property needed.>	<Any from the list of definitive context, below>	<Non-definitive information the author considers informative.>	<References to source material.>
2				
…				

Table 3-9 Dependency-Guarantee Relationship Template

The *concise* d*efinition* of the guarantee/dependency covers all aspects of what is being offered including any status or error reporting/handling e.g. “I guarantee to provide speed OR return a failure code”.

Some dependencies may not be related to a specific guarantee, and may be captured in a list of block-wide dependencies similar to the second half of the above table.

The DGR may include *definitive context* for the guarantee over and above its *concise definition*.

Some *definitive context* may be considered more significant to safety, depending upon the specific system. Variations of the above DGR table are often used in order to help clarify or capture different *definitive context*, which affects the appearance of this artefact.

The *definitive context* includes any of that listed in the table below.

List of Definitive Context	
For Guarantees (or their Relation to Dependencies)	**For Dependencies**
Any assumptions or restrictions on the usage of the guarantee. This includes operational restrictions, tolerance to environmental conditions and any behaviour or properties that are identified as needing to be considered by the system level Hazard Analysis.	Any assumptions about or limitations to what is needed by the dependency. This includes the definition of the scope of intended operating environment and its environmental conditions. (Note: any limits on behaviour or properties imposed as a result of the system level hazard analysis will manifest as requirements.)
References to definitions of terms re-used[1] within the description of the guarantee.	References to definitions of terms re-used within the description of the dependency.
Details of data provisioned by a guarantee, e.g. its latency, update rate, units, precision or the encoded identity of the transport channel(s) - as necessary to satisfy completeness.	Details of data needed by a dependency, e.g. its latency, update rate, units, precision or the encoded identity of the transport channel(s) - as necessary to satisfy completeness.
Information on any placeholders that need to be replaced (at least conceptually) to create multiple instances of DGRs. If present they make this a generic DGR template.	Information on any placeholders that need to be replaced (at least conceptually) to create multiple instances of DGRs. If present they make this a generic DGR template.
References to/identification of any dependencies that may need to be satisfied by the consumer before the guarantee is valid (counter-dependencies), for example to a: • dependency on prior initialisation • dependency on provision of resources such as an output buffer to write the data to.	References to/identification of any guarantees that may need to be provided to the supplier satisfying the dependency (counter-guarantees), for example: • to a guarantee that the supplier has previously been initialised. • to a guarantee of provision of a resource such as an output buffer for data to be written to.
Not used.	A record of the coupling between two dependencies, such as between a data specification and a communication service dependency that were separated to allow their independent satisfaction.
A reference to any list of block-wide dependencies that are related to the guarantee.	n/a

Table 3-10 Definitive Context for Dependencies and Guarantee

Traceability from the *concise definition* and the *definitive context* to the core engineering artefacts (or to analyses of them) may be recorded on the DGR.

[1] Authors are encouraged to create their own terms where others exist already.

The content of the *incidental notes* column is unspecified; DGR authors have found it useful to make notes during the work for the benefit of reviewers. The DGR consumer is not dependent on these *incidental notes*, and they do not feature in the argument.

Dependency-Guarantee Relationships are often shown diagrammatically using circles to represent the Block Safety Case Modules.

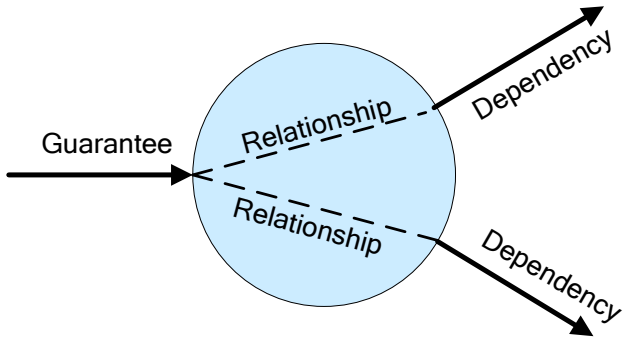


Figure 3-3 Dependency-Guarantee Relationship

	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8
Produced by			✓			✓		
Refined by						✓		
Used by			✓	✓	✓	✓	✓	✓

Table 3-11 Steps associated with the Dependency-Guarantee Relationship Artefact

3.9 Hardware Safety Case Wrapper

Any software system will be dependent on hardware on which the software executes. Where it is necessary to make a safety claim for such a dependence on hardware this can be achieved by providing a wrapper Safety Case Module that essentially supports the claim that the specific hardware is both available and sufficiently dependable. The detailed elaboration of this argument will be presented as artefacts that naturally emerge from the appropriate hardware development process.

	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8
Produced by			✓			✓		
Refined by						✓		
Used by			✓	✓	✓	✓	✓	✓

Table 3-12 Steps associated with the Hardware Safety Case Wrapper Artefact

There may be none but more likely several Hardware Safety Case Wrapper Modules in a Safety Case.

3.10 Hazard Mitigation

This artefact captures an argument with supporting evidence that claims the specified mitigation (safety-related) requirements do in fact mitigate the identified hazards. This argument may be presented in GSN.

	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8
Produced by			✓			✓		
Refined by						✓		
Used by			✓	✓	✓	✓	✓	✓

Table 3-13 Steps associated with the Hazard Mitigation Safety Case Module Artefact

3.11 Impact Assessment

This artefact captures the impact of a potential change on the Safety Case. An Impact Assessment could be presented as markups to GSN, see Figure 3-4, and/or in Tabular form, see Table 3-14.

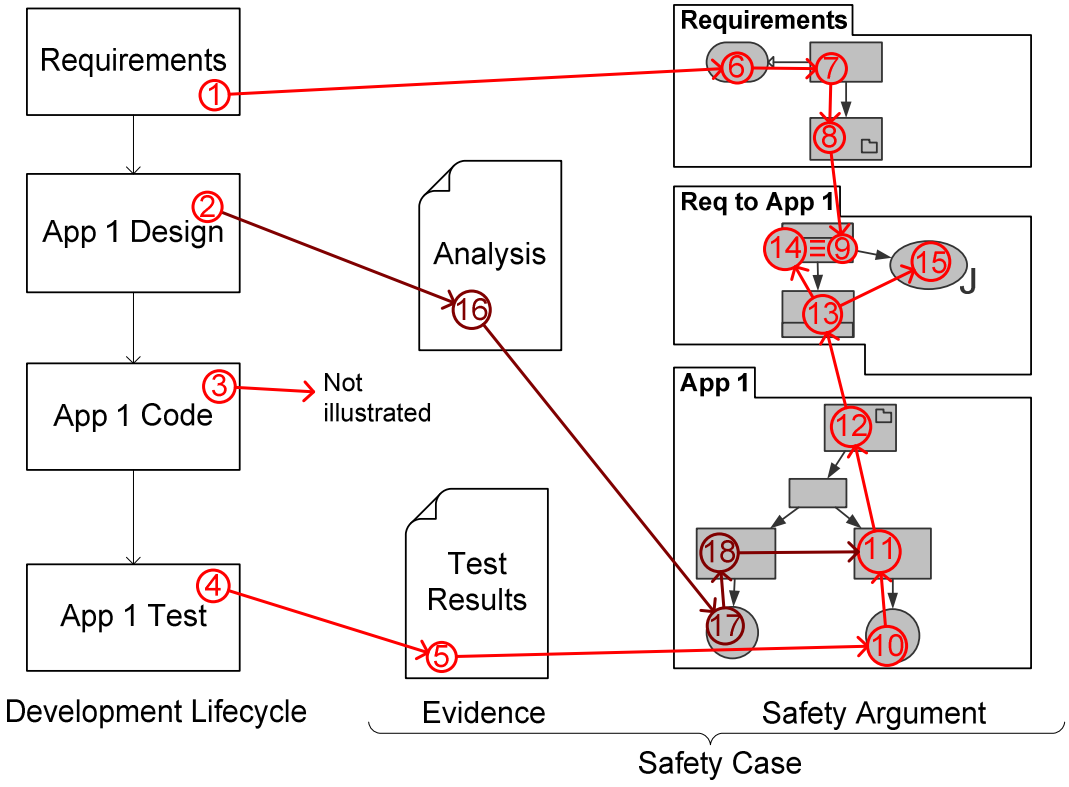


Figure 3-4: Illustration of Diagrammatic Approach to Impact Assessment

Change ID	Driving Change ID	Impacted SC Artefact			Text of SC Artefact or Artefact Description		On SC Module Boundary?	Assessed or Justification	Applied to SC?	Further Impact
		Scope (SC Module)	Type	Name	Baseline (Before)	Changed (After)				
6	1	Requir-ements	Context				No	Assessed		7
7	6	Requir-ements	Claim				No	Assessed		8
8	7	Requir-ements	Claim				Yes	Assessed		9
etc.										

Table 3-14 Example Tabular Form of Impact Assessment

Mark-Up form is essentially the Safety Case Architecture diagram in GSN format (as in Figure 3-6) highlighted to show the impact of any changes in requirements, software design or test evidence.

	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8
Produced by					✓			
Refined by								
Used by						✓		

Table 3-15 Steps associated with the Impact Assessment Artefact

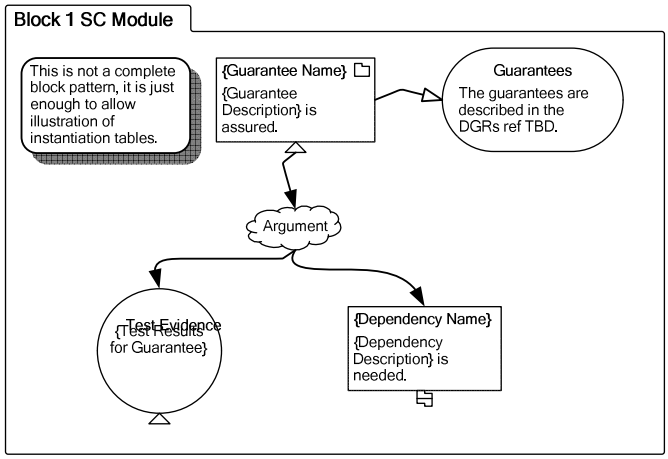
There would be an Impact Assessment artefact produced each time the Safety Case is incremented.

3.12 Instantiation Tables

There may be situations where the same rationale or argument is repeated with very small changes in the context or requirements. In this case it may be possible to create generic GSN argument along with common or generic evidence. Generic forms of Dependency-Guarantee Relationships and Dependency-Guarantee Contracts could also be constructed.

The actual form and content the generic Safety Case module will vary however the following diagram illustrates a typical example:

Artefacts being instantated



Dependency – Guarantee Relationship: {DGR Name}		
Concise Definition		**Definitive Context**
{Guarantee Description}		
N°.	**Concise Definition**	**Definitive Context**
1	{Dependency Description}	
2		
…		

This is not a complete DGR table, it is just enough to allow illustration of instantiation tables.

Corresponding Instantiation Table

{DGR Name} and {Guarantee Name}	{Guarantee Description}	{Dependency Name}	{Dependency Description}	{Test Results for Guarantee}
DGR1	Output of Speed on...	DGR1.D1	Input of Distance on...	Test Spec 5.1
		DGR1.D2	Input of Time on...	Test Spec 5.2

Figure 3-5 Instantiation Table Example

	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8
Produced by			✓			✓		
Refined by						✓		
Used by			✓	✓	✓	✓	✓	✓

Table 3-16 Steps associated with the Instantiation Tables Artefact

There may be no need Instantiation Tables artefacts in a Safety Case.

3.13 Integration SC Module

This artefact provides arguments about supporting properties of a set of interdependent modules. An Integration Module will be associated with several Block Modules.

	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8
Produced by				✓				
Refined by								
Used by								✓

Table 3-17 Steps associated with the Integration Module Artefact

There may be no need for Integration Modules in a Safety Case.

3.14 Lifecycle Plan

This artefact captures a concise but complete description of the lifecycle of the Software product. A conventional software development plan will focus on the delivery of a particular release detailing the development environment to be used, the specific requirements to be met and the means by which this will be achieved. This level of detail is not required for the Safety Case Lifecycle Plan. Rather the focus should be on the development and maintenance of the Software Product over its entire lifetime with particular emphasis on how it might be updated, adapted and modified for future use. The Lifecycle Plan will normally contain a range of scenarios which capture the potential updates to the software product and hence facilitate an analysis of the areas of the source code most susceptible to change. This information can be used to guide the design and implementation of the software in order to encapsulate the areas unlikely to change.

The Lifecycle Plan will be revisited and potentially updated (particularly for agile developments) for each increment of the Safety Case.

	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8
Produced by	✓							
Refined by		✓			✓			
Used by		✓			✓			

Table 3-18 Steps associated with the Lifecycle Plan Artefact

It would be normal to have one Lifecycle Plan in each Safety Case.

3.15 Physical Architecture

This artefact describes the physical architecture in which the Software will execute. Within the architecture elements of software that will be argued about by a single Safety Case Module are organised into Blocks. These blocks will form a part of the Safety Case Architecture. Hence the granularity of decomposition will determine the degree to which elements of the Software Safety case.

	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8
Produced by								
Refined by								
Used by		✓						

Table 3-19 Steps associated with the Physical Architecture Artefact

3.16 Safety Case Architecture

This artefact defines the architecture of the Safety Case. It should identify all Safety Case Modules that make up the Safety Case. An example of how this could be done is in section 13 of [201]. An example Safety Case Architecture is shown below:

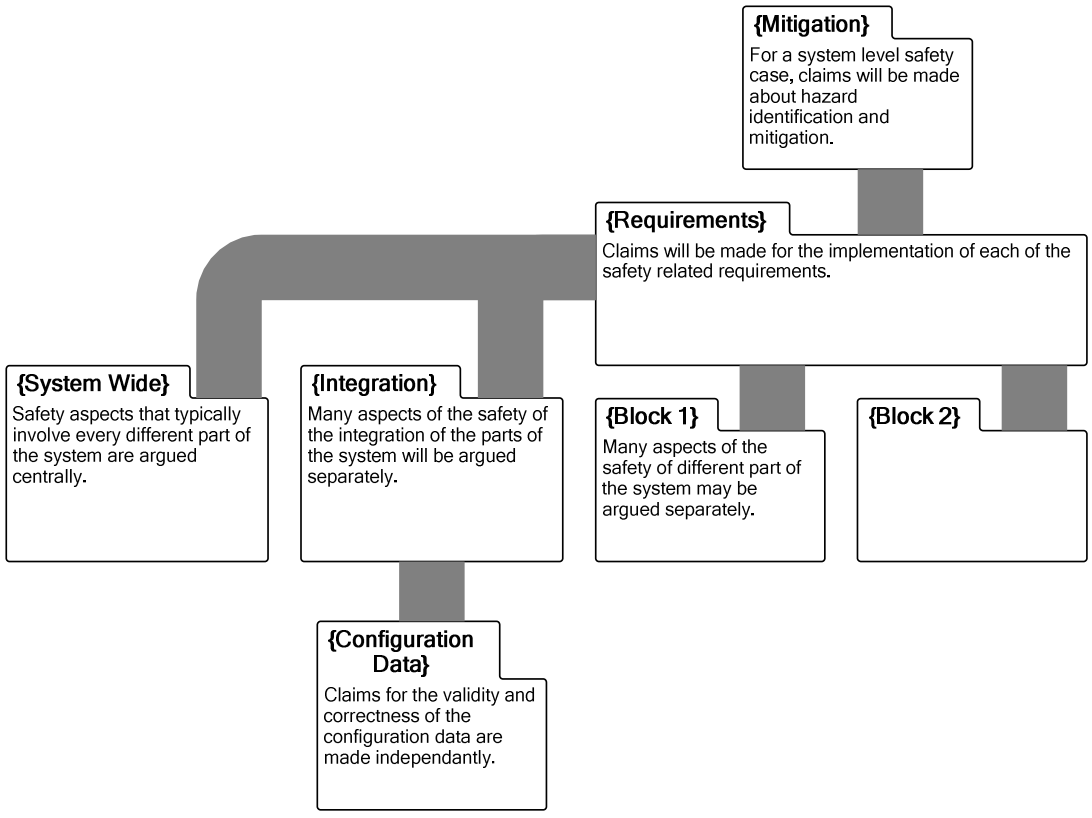


Figure 3-6 Generic Safety Case Architecture

	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8
Produced by		✓						
Refined by		✓	✓			✓		
Used by		✓	✓	✓	✓	✓	✓	✓

Table 3-20 Steps associated with the Safety Case Context Artefact

There will only be one Safety Case Architecture artefact per Safety Case.

3.17 Safety Case Module Context

Context defines the conditions in which a Safety Case Module can be relied on and expects to be supported. It should capture all elements of context that elaborate, restrict or constrain the validity of the Safety Case Module.

	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8
Produced by			✓			✓		
Refined by						✓		
Used by			✓	✓	✓	✓	✓	✓

Table 3-21 Steps associated with the Safety Case Context Artefact

There will be a set of context for each Safety Case Module.

3.18 Safety Case Report

The Safety Case Report describes the Safety Case Module or the Top Level Safety Case and makes the formal claim of the safety of the software. It collates or references all the artefacts the Safety Case Module contains.

The typical table of contents for a Safety Case Module Report may appear as follows:

	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8
Produced by				✓				
Refined by							✓	
Used by								✓

Table 3-22 Steps associated with the Safety Case Report Artefact

There would be one issue of the Safety Case report produced for each issue of a Safety Case Modules it describes.

3.19 Software Safety-Related Requirements SC Module

Any safety-related system will have a set of defined safety requirements that have come from contractual requirements, standards and legislation or are the results of Hazard Analysis. Addressing the safety requirements is often the starting point for a Modular Safety Case, the argument being that if the system fulfils all the Software Safety-Related Requirements (SSRs) it can be considered "safe" (in the defined context of usage). This artefact captures all such SSRs and provides an argument that each is sufficiently analysed and directed to a Block or Blocks that will claim to satisfy it. These safety requirements provide the goals for the rest of the Safety Case and it may be convenient to group the requirements and associated completeness and requirements analysis arguments together in a SSR Safety Case Module.

	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8
Produced by			✓			✓		
Refined by						✓		
Used by			✓	✓	✓	✓	✓	✓

Table 3-23 Steps associated with the Safety-Related Requirements Module Artefact

There will be one or more Safety-Related Requirements Modules for each Safety Case.

3.20 System Wide Issues SC Module

In any system there are inevitably certain aspects of behaviour or performance that cannot be ensured by a single Block or argued by a single Safety Case Module - they have to be argued at a system-wide level. End to end latency is an example of this where several Blocks may contribute to latency but none has total control over it. So an argument is made at System Wide level, supported by test / analysis evidence at the system level such as actual end to end timing tests.

	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8
Produced by			✓			✓		
Refined by						✓		
Used by			✓	✓	✓	✓	✓	✓

Table 3-24 Steps associated with the System Wide Issues Artefact

There may be none but more likely there would be several System Wide Issues Modules in each Safety Case.