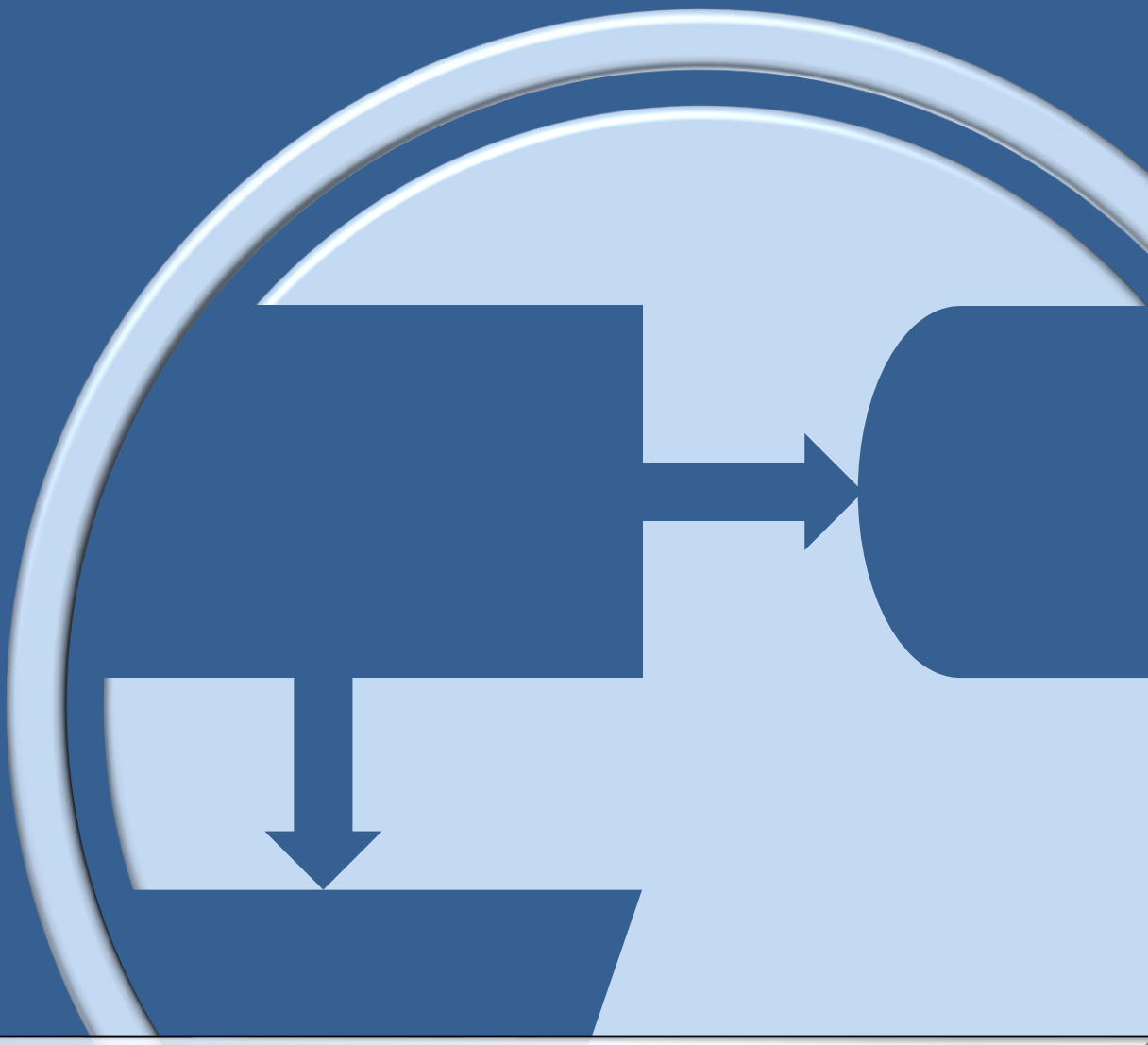


Goal Structuring Notation Community Standard

Version 3

The Assurance Case Working Group (ACWG)

SCSC-141C

DECLARATION

Some of the material presented in this Standard has previously been published in the following:

- The Six-Step Method for Developing Goal Structures (Section 2:3) in Kelly, T. 'Arguing Safety: A Systematic Approach to Managing Safety Cases', D. Phil Thesis, University of York (1988).
- The review process presented in Section 2:7.4 in Kelly, T. 'Reviewing Assurance Arguments – A Step-by-Step Approach', in Proceedings of the Workshop on Assurance Cases for Security – The Metrics Challenge, Dependable Systems and Networks (DSN), July 2007

The material is reproduced here with the permission of the original author, who retains rights to the material.

DISCLAIMER

While the authors and the publishers have used reasonable endeavours to ensure that the information and guidance given in this work is correct, all parties must rely on their own skill and judgement when making use of this work and obtain professional or specialist advice before taking, or refraining from, any action on the basis of the content of this work. Neither the authors nor the publishers make any representations or warranties of any kind, express or implied, about the completeness, accuracy, reliability, suitability or availability with respect to such information and guidance for any purpose, and they will not be liable for any loss or damage including without limitation, indirect or consequential loss or damage, or any loss or damage whatsoever (including as a result of negligence) arising out of, or in connection with, the use of this work. The views and opinions expressed in this publication are those of the authors and do not necessarily reflect those of their employers, the SCSC[1] or other organisations.

[1] SCSC : Safety-Critical Systems Club C.I.C. A Community Interest Company registered in England (Company number 13084663)

FOREWORD

This Standard has two intended functions. Firstly, it seeks to provide a comprehensive, authoritative definition of the Goal Structuring Notation (GSN). Secondly, it aims to provide clear guidance on current best practice in the use of the notation for those concerned with the development and evaluation of engineering arguments [2] – argument owners, readers, authors and approvers.

The first version of the Standard was developed by means of a consensus process involving GSN users from both academia and industry, between 2007 and 2011. It is thus a standard written for the community, by the community. Subsequent versions have continued to be developed by the GSN user community to address comments and suggestions from users based on their experiences using the notation in the intervening period, whilst also addressing developments in the field of structured arguments. The document history outlines the recent history of the collaboration, and a list of contributors to the Standard is provided on page 5.

Some of the contributors to this standard were involved in the Object Management Group's (OMG) development of the Structured Assurance Case Metamodel (SACM) [7]. A meta-model of GSN, showing the relationship to SACM, can be found at scsc.uk/gsn

LICENSE

[2] It is noted that structured arguments can be utilized in many contexts, but the focus of this standard is when those arguments are applied in engineering disciplines.

ACCESS

This standard can be accessed for free at http://scsc.uk/SCSC-141C

DOCUMENT HISTORY

Version	Issued to	Date	Purpose
1	User Community	16th November 2011	For use
2	User Community	January 2018	For Use
3	User Community	May 2021	For Use

Change History

Version 2 collated comments resulting from use of Version 1. These were considered by the review group and a series of clarifications and minor corrections incorporated.

Version 3 has addressed substantial topics such as development of the notation for modular arguments and associated guidance, introduction of Argument Claim Points and provisions for dialectic arguments.

Future Development

The ACWG intends to continue to maintain the standard in line with feedback from the user community and to reflect developments in approach to arguments.

Users of the standard are invited to continue to provide feedback on the use of the standard, and to suggest areas for further development. See www.scsc.uk/gc for further details.

CONTRIBUTORS

The following individuals and organizations are acknowledged for their contribution to and support for the GSN standardization process. Contributors have not necessarily contributed to all versions and the resultant standard does not necessarily entirely reflect the views of those acknowledged here.

Individual Contributors

Katrina Attwood
Mark Carter
Paul Chinneck
Martyn Clarke
George Cleland
Mark Coates
Trevor Cockram
George Despotou
Luke Emmet
Jane Fenn
Ben Gorry
Ibrahim Habli
Christopher Hall
Andrew Harrison
Richard Hawkins
Pete Hutchison
Andrew Jackson
Tim Kelly
Peter Littlejohns
Lisa Logan
Paul Mayo
Yvonne Oakshott
Ron Pierce
Clive Pygott
Graeme Scott
Mick Warren
Ran Wei
Andy Williams
Phil Williams

Contributing Organisations

AACE
Adelard
Altran
BAE Systems
Blackberry-QNX
Columbus Computing
CSE International
Dalian University of Technology
Engineer for Safety
General Dynamics UK
Leonardo
LR Rail
New Technologies
RINA
RPS Group
SafeEng
Selex-Galileo
Thales
UK Ministry of Defence
University of York

TABLE OF CONTENTS

INTRODUCTION TO THE STANDARD

The purpose of this Standard is to define the Goal Structuring Notation (GSN) and to provide guidance on its usage. GSN is a graphical argument notation that can be used to document explicitly the individual elements of any argument (claims, evidence and contextual information) and the relationships that exist between these elements (i.e. how claims are supported by other claims, and ultimately by evidence) and the context that is defined for the argument. GSN is a generic argument structuring language, which can be used to document arguments in any domain. In this Standard, examples from the assurance case domain are used, such as safety and security arguments.

The Standard has four parts, as follows:

- Part 0: Introduction And Concepts (informative). This part provides an overview of the concepts of GSN and its role in communicating arguments. It can be used as a standalone introduction to GSN.
- Part 1: Definition Of The Goal Structuring Notation (normative). This part provides a normative definition of the syntax of GSN, including its visual syntax and the semantics of the notation, clarifying the meanings of standard GSN structures. It includes sections that define the syntax and semantics of extensions that have been made to GSN, for example those made to enable GSN to describe generic argument patterns and modular argument structures.
- Part 2: Guidance On The Development And Evaluation Of Goal Structures (informative). This part provides informative guidance on the effective use of GSN to create and evaluate structured arguments.
- In addition, there is a web-based repository of GSN resources. The repository will provide additional informative guidance on the use of GSN, including further examples of goal structures and catalogues of existing argument patterns. These resources are at scsc.uk/gsn and are under continuous development.

Part 0: INTRODUCTION AND CONCEPTS

0:1 Introductory

0:1.1 This part of the Standard provides sufficient information about GSN to enable a novice user to read and understand a goal structure represented using the notation without recourse to the remainder of the Standard.

0:1.2 GSN is a generic language for structuring arguments. As such, it can be used to present arguments in any domain. Examples in this Standard are taken from the domain of assurance cases.

0:2 Use of Arguments in Assurance Cases

0:2.1 The concept of assurance cases has long been established in the assurance domain where for many industries the development, review and acceptance of an assurance case forms a key element of safety assurance processes.

0:2.2 An assurance case can be defined as:

> A reasoned and compelling argument, supported by a body of evidence, that a system, service or organisation will operate as intended for a defined application in a defined environment.

0:2.3 In practice, an assurance case will have a particular focus. For example, a safety case will demonstrate that a given system is acceptably safe in a given context, while a security case will justify the security properties of a system.

0:2.4 In order that assurance cases can be developed, discussed, challenged, presented and reviewed amongst stakeholders, and maintained throughout the product lifecycle, it is necessary for them to be documented clearly. The documented argument of the assurance case should be structured to be comprehensible to all assurance-case stakeholders. It should also be clear how the evidence is being asserted to support this argument.

0:3 What is an Argument?

0:3.1 In the sense used in assurance cases, an argument is defined as a connected series of claims intended to establish an overall claim. In attempting to persuade others of the truth of an overall claim, supporting claims are used. These claims may themselves need further support. Ultimately, claims should be supported by reference

to evidence. This gives rise to a hierarchy of claims (representing a logical chain of reasoning supported by evidence) by which an argument is established.

0:3.2 At the heart of GSN is the explicit documentation of the hierarchy of claims and evidence. The top goal presents the overall claim asserted by the author and it is up to the reader to determine their belief that it is adequately supported. This standard does not assert that an argument presented in GSN is necessarily sufficient to support the overall claim, rather the use of GSN enables an author to clearly present the basis of their support for that claim and the reader to comprehend and challenge that basis and thereby evaluate the confidence in support of the claim. It is considered essential good practice to evaluate the goal structure during authoring, approval and maintenance with a critical view to remove potential bias. The use of a dialectic[3] approach to evaluation is particularly recommended. The key elements of the notation are explained in Section 0:4.

0:4 The Goal Structuring Notation (GSN)

0:4.1 GSN is a graphical argument notation which can be used to document explicitly the elements and structure of an argument and the argument's relationship to evidence. In GSN, the claims of the argument are documented as goals and items of evidence are cited in solutions. The relationships represented in GSN are:

- The premise-conclusion relationship between supporting goals and their parent goal;
- The support that solutions provide for goals; and
- The relationship between the argument and the context in which it is stated.

0:4.2 The purpose of GSN is to document how claims (conclusions, represented in GSN as goals) are said to be supported by sub-claims (premises, also represented in GSN as goals). Figure 0:4-1 shows an example goal in GSN:

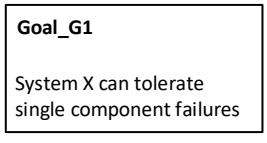


Figure 0:4-1 An Example Goal

[3] 'dialectic' is defined by the oxford English dictionary as "Logic, reasoning; critical investigation of truth through reasoned argument, often spec. by means of dialogue or discussion."

0:4.3 Where evidence is asserted to support the truth of the claim, this can be documented by providing a solution in GSN. Figure 0:4-2 shows an example solution (reference to evidence) in GSN:

Figure 0:4-2 An Example Solution

0:4.4 When documenting how claims are said to be supported by sub-claims, it can be useful to document the reasoning step – i.e. the nature of the argument that connects the claim to its sub-claims. This is done in GSN by documenting the strategy of the argument which links goals. Figure 0:4-3 shows an example strategy in GSN:

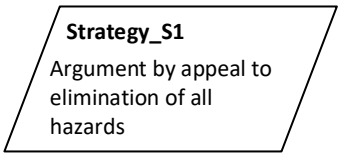


Figure 0:4-3 An Example Strategy

0:4.5 When documenting a GSN goal or strategy it can also be important to capture the context in which the claim or reasoning step should be interpreted. This is done in GSN by documenting context. Figure 0:4-4 shows an example context in GSN:

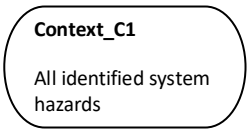


Figure 0:4-4 An Example Context

0:4.6 Some claims and argument strategies are expressed in the context of assumptions. These assumptions must be valid for the claim or the strategy to be valid. Assumptions can be documented explicitly in GSN using the assumption element. An example of an assumption can be seen in Figure 0:4-5; Strategy S1 is stated in the context of an assumption that all credible hazards have been identified.

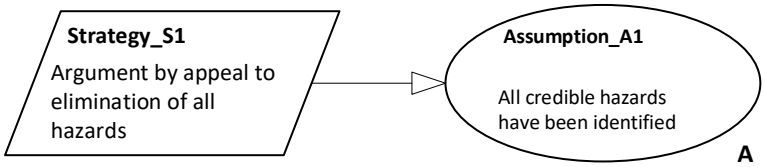


Figure 0:4-5 An Assumption stated about a Strategy

0:4.7 Argument authors may feel the need to justify a particular claim or argument strategy, to provide some explanation as to why they consider it acceptable. This is achieved in GSN by the use of the justification element. An example of a justification can be seen in Figure 0:4-6; the argument author justifies the use of an argument approach using Safety Integrity Levels (SILs) by asserting that SIL apportionment is recognised by an appropriate safety standard.

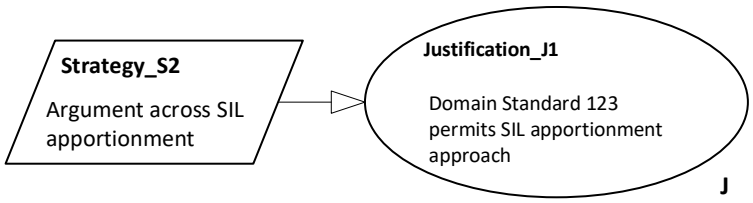


Figure 0:4-6 An Example Justification

0:4.8 Goals, solutions, strategies, contexts, assumptions and justifications form the principal elements of GSN. (A full description of all GSN element-types is provided in Part 1: below).

0:4.9 GSN provides two types of linkage between elements: SupportedBy and InContextOf. SupportedBy relationships – represented by lines with solid arrowheads – indicate inferential or evidential relationships between elements. InContextOf relationships – represented as lines with hollow arrowheads – declare contextual relationships.

0:4.10 When the elements of GSN are connected together, they are said to form a ‘goal structure’. Figure 0:4-7 shows an example goal structure.

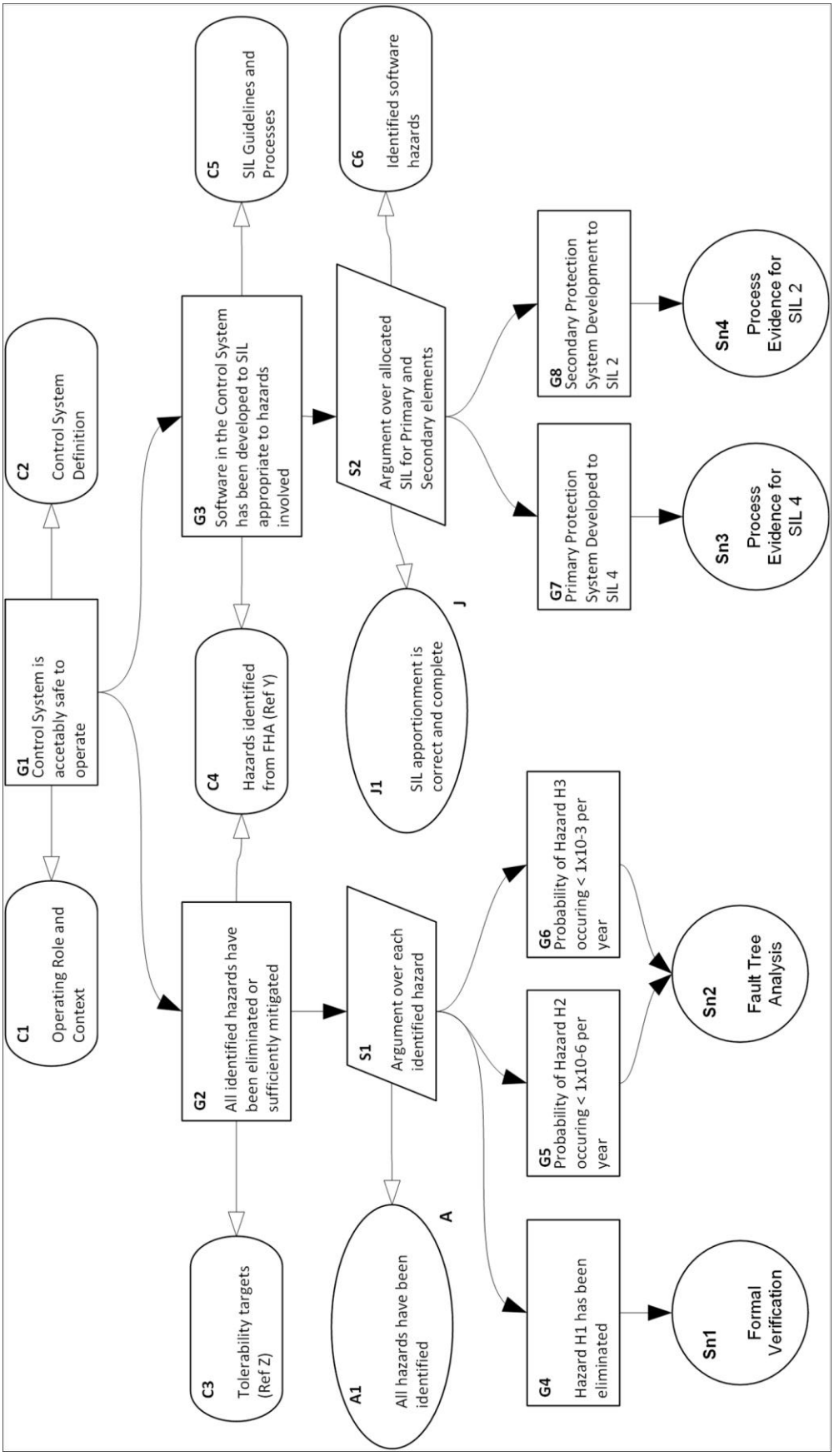


Figure 0:4-7 An Example Goal Structure

0:4.11 Goal structures document the asserted chain of reasoning in the argument (through the visible decomposition of claimed goals and the description of argument strategies) and indicate how this argument is supported by evidence (through solutions). The goal structures also clearly document the context in which the claims of the argument are being put forward.

0:4.12 It is important to recognise that GSN simply provides a means of documenting an asserted argument. The use of GSN itself does not establish the truth of that argument.

0:4.13 The key benefit from using an explicit approach such as GSN to develop and document the arguments of any assurance case is that it can improve comprehension amongst the key stakeholders (e.g. in assurance cases, system developers, engineers, independent assessors and certification authorities). This in turn can improve the quality of the debate and the time taken to reach agreement on the argument approaches being adopted. For example, using the goal structure provided in Figure 0:4-7, it would be reasonable to question whether the allocation of SIL 4 to the primary protection system and SIL 2 to the secondary protection system had been adequately demonstrated to be appropriate to the hazards involved. This discussion could lead to a requirement for a SIL allocation justification.

Part 1: DEFINITION OF THE GOAL STRUCTURING NOTATION

1:1 Introductory

1:1.1 This part of the Standard provides a normative definition of the Goal Structuring Notation: it describes permitted structures and formulations in GSN. Note that it does not prescribe good practice – guidance on that is provided in Part 2:. GSN defines elements, the allowable relationships between these elements and the acceptable language of the text within these elements. Each element comprises a graphical symbol and a textual statement.

1:1.2 Elements are composed into arguments by providing text within the elements and relating them to other elements. GSN considers that all arguments are contained in an argument module. In core GSN this module is typically implicit.

1:1.3 GSN was originated at the University of York in the early 1990s as part of the ASAM-II project [1], and has undergone significant development and refinement since then. The early development of GSN was heavily influenced by Toulmin's work on arguments [2] and emerging goal-based approaches to requirements engineering, such as KAOS [3].

1:1.4 The core elements of the notation are introduced in Section 1:2. Section 1:2.2 describes the interpretation of permitted combinations of these elements. Section 1:2.3 defines the language used within the symbols. Extensions to the core GSN to support the development of generic argument patterns and modularised arguments are defined in sections 1:3 and 1:4 respectively.

1:2 Core GSN

1:2.1 Notation

1:2.1.1 GSN defines the following core elements:

- Goal
- Strategy
- Solution
- Context
- Assumption
- Justification

1:2.1.2 Each element contains an element identifier. The identifier shall identify the element uniquely within an argument module. The element identifier is represented

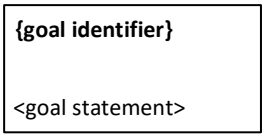
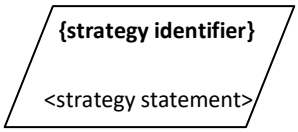
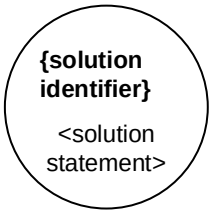
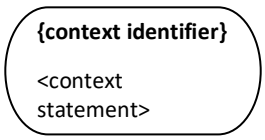
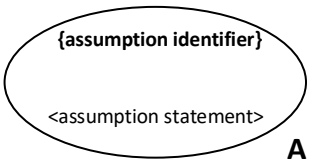
here within curly brackets, as with other aspects identifying elements, such as the module identifier. Descriptive text which will be replaced in its entirety in goal structures is captured in angle brackets.

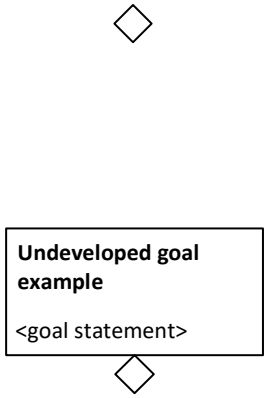
1:2.1.3 These core elements are linked using the following types of relationships:

- SupportedBy
- InContextOf.

1:2.1.4 Table 1:2-1 provides the definition and rendering of these elements. GSN relationships are defined in para 1:2.1.5 below. The meanings of structures combining these relationships are further explained in Section 1:2.2.

Table 1:2-1 Core GSN Elements

GSN Element Rendering	Definition
	A **goal**, rendered as a rectangle, presents a claim forming part of the argument.
	A **strategy**, rendered as a parallelogram, describes the inference that exists between a goal and its supporting goal(s).
	A **solution**, rendered as a circle, presents a reference to an evidence item.
	A **context**, rendered as shown left, presents a contextual artefact. This can be a reference to contextual information, or a statement.
	An **assumption**, rendered as an oval with the letter 'A' at the top- or bottom-right, presents an intentionally unsubstantiated statement.

GSN Element Rendering	Definition
	A **justification**, rendered as an oval with the letter 'J' at the top- or bottom-right, presents a statement of rationale.
	**Undeveloped element decorator**, rendered as a hollow diamond applied to the bottom centre of an element, indicates that a line of argument has not been developed. It can apply to goals (as below) and strategies. For example, an **undeveloped goal**, rendered as a rectangle with the hollow-diamond undeveloped element decorator at the centre-bottom, presents a claim which is intentionally left undeveloped in the argument.

1:2.1.5 The core GSN elements defined here are intended to be combined to represent logical structures, known as 'goal structures'. GSN provides two core types of relationship between elements, as indicated in Table 1:2-2. These declare a relationship between a source element and a target element. The arrow points to the target.

Table 1:2-2 Core GSN Relationships

GSN Relationship Rendering	Definition
	**SupportedBy**, rendered as a line with a solid arrowhead, allows support relationships between elements to be documented. Permitted 'supported by' connections are: goal-to-goal, goal-to-strategy, goal-to-solution, strategy to goal.
	**InContextOf**, rendered as a line with a hollow arrowhead, declares a contextual relationship. Permitted 'in context of' connections are: goal-to-context, goal-to-assumption, goal-to-justification, strategy-to-context, strategy-to-assumption and strategy-to-justification.

1:2.2 Notation Interpretation

1:2.2.1 This section introduces the rules which govern the relationships between graphical elements of the GSN are introduced.

1:2.2.2 A GSN goal structure is a directed acyclic graph. This means that the graph does not allow loops, although one element can have multiple parents and children. SupportedBy relationships shall not be constructed so as to directly or indirectly allow a goal to support itself. Similarly, InContextOf relationships shall not be constructed so as to directly or indirectly allow a goal to provide its own context.

1:2.2.3 Figure 1:2-1 shows the most basic relationship represented in goal structures – inference between goals:

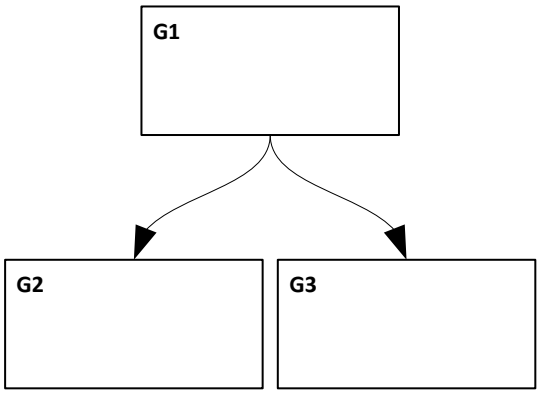


Figure 1:2-1 Supporting Goals with Other Goals

1:2.2.4 This specific structure illustrates the inferential relationship between the conclusion presented in G1 and its premises presented in supporting goals G2 and G3. It asserts that if the claims presented in Goals G2 and G3 are true, this is sufficient to establish that the claim in Goal G1 is true. G1 may be referred to as the parent goal, whilst G2 and G3 would commonly be referred to as ‘supporting goals’, ‘sub-goals’ or ‘child goals’ of G1. One or more supporting goals may be declared for a given goal. It is noted that the inferential relationship between the goal and its supporting goals is provided by the indivisible combination of the two ‘SupportedBy’ relationships.

1:2.2.5 The structure shown in Figure 1:2-2 also asserts that if the claims presented in Goals G2 and G3 are true, this is sufficient to establish that the claim in Goal G1 is true. However, a GSN strategy (S1) has been added to the diagram to describe the inference which is asserted as existing between goals G2 and G3 and the parent goal

G1. The text of strategy S1 would explain to the reader how Goals G2 and G3 support Goal G1.

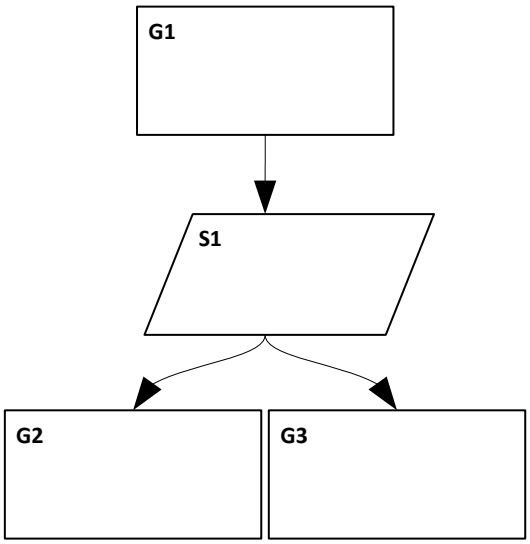


Figure 1:2-2 Adding Strategy

1:2.2.6 In some cases, more than one argument approach may be adopted in support of a parent goal. Figure 1:2-3 represents a relationship of this type, by which the separate contributions made by each of the goal groupings (G2, G3) and (G4, G5) to the argument supporting goal G1 are made explicit in strategies S1 and S2 respectively. Both lines of argument are required to support goal G1. Strategy S1 is a description of the argument that is being asserted to relate the goals G2 and G3 to the parent G1. Strategy S2 describes the argument relating G4 and G5 to G1. It is noted that the inferential relationship between the goal and its supporting goals is provided by the indivisible combination of the three ‘SupportedBy’ relationships and is the same as that shown in Figure 1:2-2.

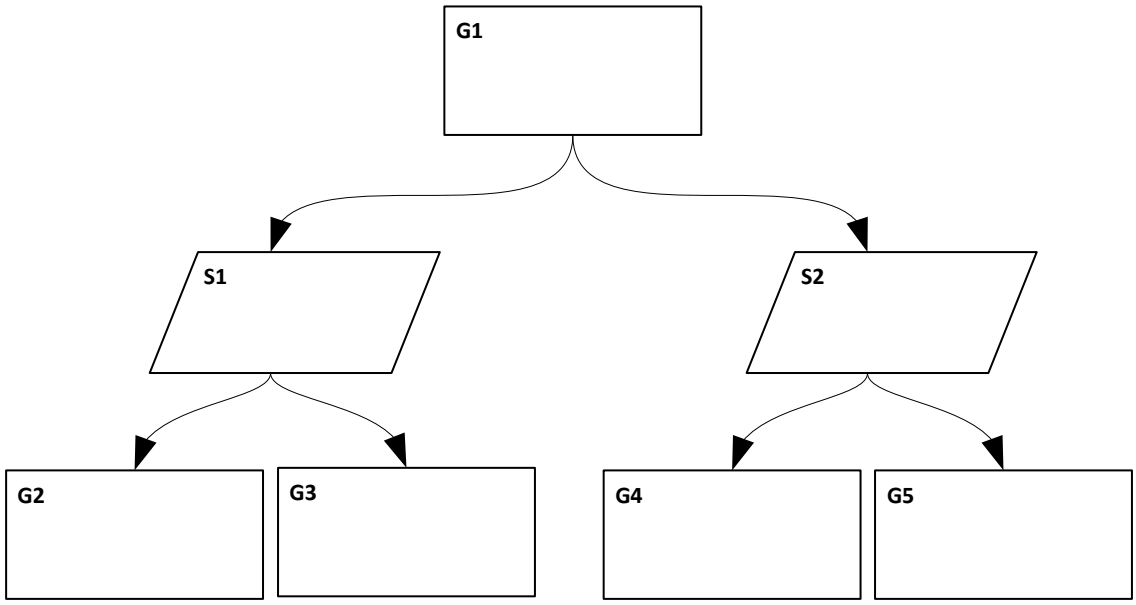


Figure 1:2-3 Multiple Strategies

1:2.2.7 Figure 1:2-4 represents the use of a reference to an evidence item to support a claim.

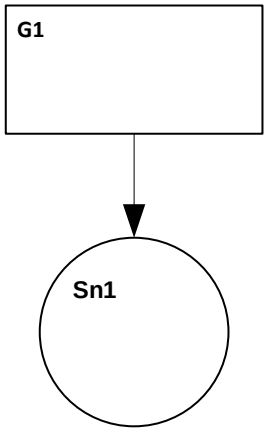


Figure 1:2-4 Providing Solutions

1:2.2.8 This structure represents an evidential relationship that asserts that the evidence referred to in the solution (Sn1) is sufficient to establish the truth of the claim made in the goal (G1).

1:2.2.9 As with the use of multiple argument approaches to support a claim demonstrated in Figure 1:2-3, there may be situations in which the existence of multiple evidence items is invoked in support of a claim. In cases of this kind, multiple solutions will be presented in the goal structure. Figure 1:2-5 represents an evidential relationship that asserts that the evidence referred to in Solutions Sn1 and Sn2 is

sufficient to establish the truth of the claim made in goal G1. It is noted that the evidential relationship between the goal and its supporting evidence is provided by the indivisible combination of the two ‘SupportedBy’ relationships.

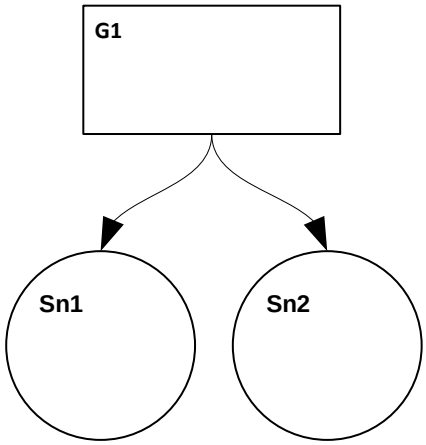


Figure 1:2-5 Multiple Solutions

1:2.2.10 Claims can only be asserted to be true in a specified context. Context elements can be used in GSN to make this relationship clear. Figure 1:2-6 shows the addition of context to a goal. The context is used to declare supplementary information related to the claim made in goal G1.

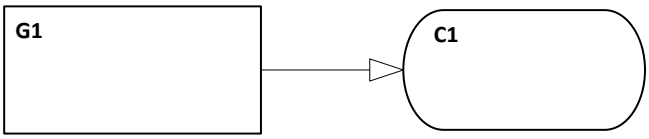


Figure 1:2-6 Adding a Context to a Goal

1:2.2.11 Where used, contexts define or constrain the scope over which the claim is made. Since a contextual statement makes an assertion in the argument structure, nothing in the supporting argument for the goal to which the context is applied should contradict or undermine the relationship between the goal and the context. Context is taken to be connected to the entirety of the argument supporting the referenced element. Therefore, it is not necessary to restate the context in the supporting argument.

1:2.2.12 An assumption applied to a goal declares an assumption made in stating the claim. The meaning of the structure in Figure 1:2-7 is that the claim in goal G1 is asserted in a context where the assumption A1 is true:

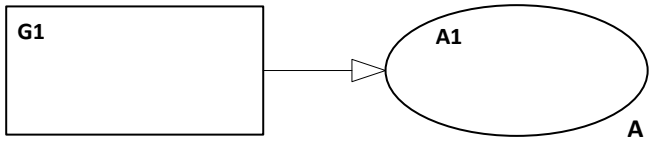


Figure 1:2-7 Adding an Assumption to a Goal

1:2.2.13 An assumption is an unsubstantiated statement. Having connected an assumption to a goal G1, the assumption is taken to be connected to the entirety of the relevant argument. Therefore, it is not necessary to restate the assumption in the relevant argument.

1:2.2.14 Figure 1:2-8 shows the connection of a justification to a goal. A justification does not alter the meaning of the claim made in the goal, but provides rationale for its inclusion or its phrasing. Unlike assumptions, justifications are not taken to be connected to the entirety of the argument supporting the referenced goal. They are local to the element to which they are linked. Should an equivalent justification be required elsewhere in the argument, it will need to be re-stated or re-linked.

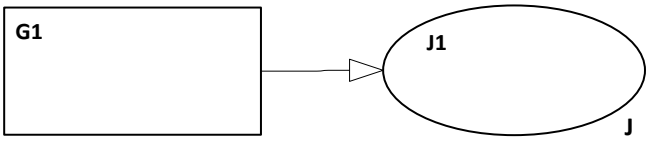


Figure 1:2-8 Adding a Justification to a Goal

1:2.2.15 A context may also be applied to a strategy to declare supplementary information related to the explanation provided in the strategy or to provide a definition or an explanation of terms used in the strategy. Figure 1:2-9 shows the addition of a context to a strategy:

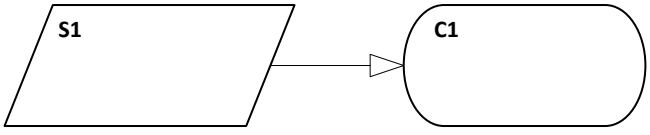


Figure 1:2-9 Adding a Context to a Strategy

1:2.2.16 As before, since a contextual statement makes an assertion in the argument structure, nothing in the supporting argument deriving from the strategy to which the context is applied should contradict or undermine the relationship between the strategy and the context. Context is taken to be connected to the entirety of the

argument supporting the referenced element. Therefore, it is not necessary to restate the context in the supporting argument.

1:2.2.17 An assumption applied to a strategy declares an assumption in how the supporting goals support the parent goal. In the structure presented in Figure 1:2-10, in declaring that the supporting goals introduced by strategy S1 are sufficient to support the parent goal, assumption A1 is taken to be true. Having connected an assumption to a strategy S1, the assumption is taken to be connected to the entirety of the argument resulting from S1. Therefore, it is not necessary to restate the assumption in the supporting argument.

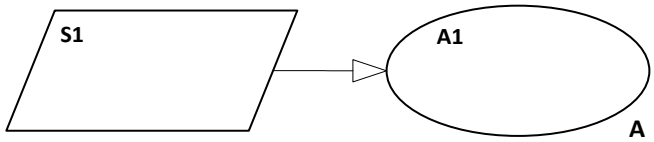


Figure 1:2-10 Adding an Assumption to a Strategy

1:2.2.18 A justification can also be connected to a strategy, to provide backing for the argument described by the strategy. Figure 1:2-11 shows the addition of a justification to a strategy:

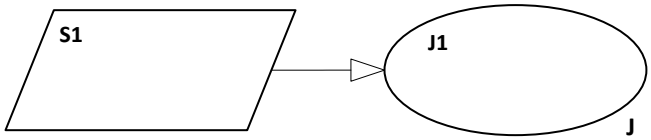


Figure 1:2-11 Adding a Justification to a Strategy

1:2.2.19 A justification applies to the element to which it is connected. Should an equivalent justification be required elsewhere in the argument, it will need to be re-stated or re-linked.

1:2.2.20 GSN structures can become large and it is often convenient to illustrate fragments of the argument structure in separate diagrams. To be able to convey that the argument continues in, or is a continuation from, a separate diagram an optional ‘off diagram’ decorator may be applied to elements at the top or bottom of each diagram. Typically these will be goal or strategy elements, and where used the element at the bottom of a diagram should be repeated as an exact copy at the top of the related diagram. The preferred off-diagram decorator is a shaded triangle spanning nominally one-third of the edge in the direction of the off-diagram link.

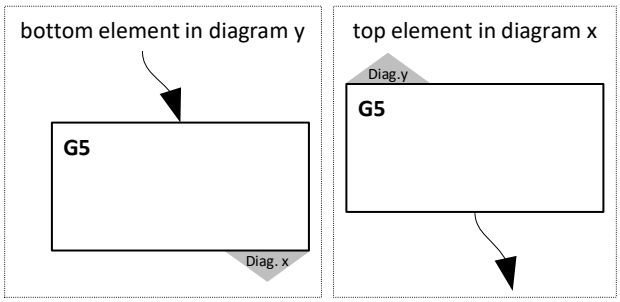


Figure 1:2-12 'Off-Diagram' Decorators

1:2.3 The Language of Goal Structures

1:2.3.1 This section presents a series of simple rules which govern the grammatical structure of statements used in GSN elements.

1:2.3.2 GSN goals document the claims made in the argument (i.e. premises and conclusions). Each goal shall contain a single goal statement, expressed as a proposition in the form of a noun-phrase + verb-phrase sentence.

1:2.3.3 GSN strategy statements describe the reasoning that connects parent goals and their supporting goals, but the core claims and the structure connecting those claims remain unchanged. Strategy statements contain a brief description of the argument approach.

1:2.3.4 GSN solutions make no claim, but are simply references to evidence items that provide support for a particular claim. They shall therefore be stated as noun-phrases.

1:2.3.5 Two kinds of GSN context statement exist. Where a context statement is a reference to an artefact of some kind, which informs the reasoning step, the context statement shall be expressed as a noun-phrase. Where a context statement draws attention to explanatory contextual information (such as the definition of some term), this information shall be stated briefly using complete sentences of a noun-phrase + verb-phrase structure.

1:2.3.6 GSN assumptions and justifications provide additional information necessary for the correct understanding of the argument. This information is stated as fully as necessary, using complete sentences in the form noun phrase + verb phrase.

1:3 Argument Pattern Extension

1:3.1 Introductory

1:3.1.1 In order to represent patterns of argument rather than merely argument instances, GSN has been extended to support structural and element abstraction.

1:3.1.2 Note that the extensions to core GSN presented in sections 1:3.2 and 1:3.3 below are intended for the representation of abstract argument patterns. Patterns should be declared in a GSN pattern definition as described in section 1:3.4.

1:3.1.3 In cases where the elements defined in the following sections are used in the development of instantiations of the patterns to produce individual assurance arguments, it is important to ensure that they are all removed, or instantiated, in the final, delivered, version of the argument. By exception, a final, delivered, version of the argument may be provided in a form that includes instantiable elements together with instantiation data as defined in section 1:3.5.

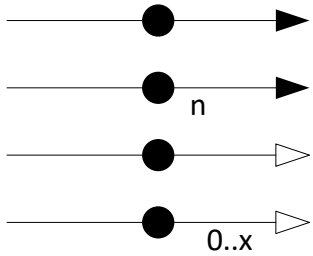
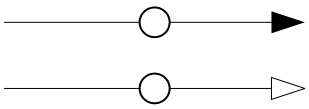
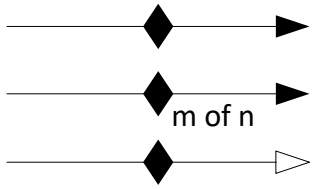
1:3.2 Structural Abstraction in GSN

1:3.2.1 This section describes the extensions to GSN defined in order to support two aspects of structural abstraction:

- **Multiplicity** – generalised n-ary relationships between GSN elements;
- **Optionality** – optional and alternative relationships between GSN elements.

1:3.2.2 Table 1:3-1 illustrates the extensions made to GSN to facilitate the representation of multiplicity. These symbols are defined for use as decorators on all existing GSN relation types. Multiplicity symbols can be used to describe how many instances of one element-type relate to another element.

Table 1:3-1 GSN Extensions for Structural Abstraction

GSN Relationship Rendering	Definition
	A solid ball is the symbol for multiple instantiations. The optional label next to the ball indicates the cardinality of the relationship. It can be expressed as an instantiable parameter relevant to the argument. If no label is included then the cardinality can be any value from one upwards. If cardinality from zero onwards is required this should be explicitly declared e.g. 0..x declares that there may be zero to x branches (inclusive). It could also be written as 0≤n≤x.
	A hollow ball indicates 'optional' instantiation, Optional instantiation means that the relationship and the argument below may or may not be instantiated.
	A solid diamond is the symbol for Choice. The optional label next to the diamond indicates the cardinality of the relationship. It can be expressed as an instantiable parameter relevant to the argument. If no label is included then the cardinality can be any value from one to the number of supporting elements.

1:3.2.3 The extension to GSN shown in Figure 1:3-1 shows the representation of structural choice A GSN choice can be used to denote possible alternatives in satisfying a relationship. In Figure 1:3-1, one goal can be supported by any one of three possible supporting goals.

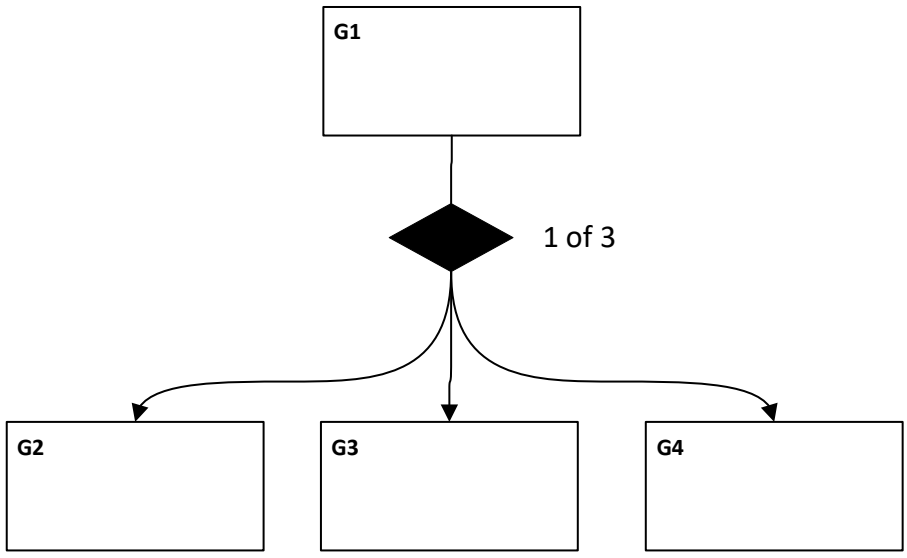


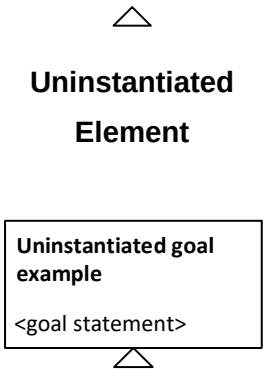
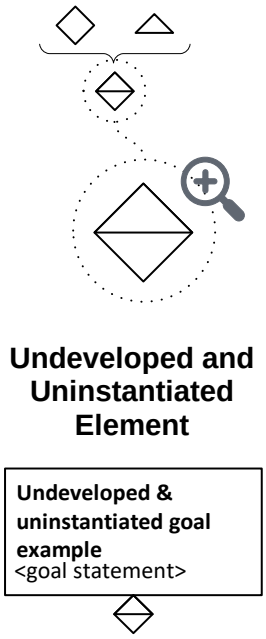
Figure 1:3-1 GSN Choice Element

1:3.2.4 Further guidance on the use of structural abstraction in GSN is provided in Section 2:8 below.

1:3.3 Element Abstraction in GSN

1:3.3.1 Table 1:3-2 illustrates extensions to GSN to enable the representation of abstract elements:

Table 1:3-2 GSN Extensions for Element Abstraction

GSN Element Rendering	Definition
 **Uninstantiated Element**	This decorator denotes that the attached element remains to be instantiated, i.e. at some later stage the 'abstract' element needs to be **replaced** (instantiated) with a more concrete instance. This decorator can be applied to any GSN element type, and should be applied to the bottom centre of the element. Example of an uninstantiated goal, demonstrating the application of the decorator. The token to be instantiated is contained within curly brackets.
 **Undeveloped and Uninstantiated Element**	Decorators can be overlaid to denote that the attached element requires both further development and instantiation. The 'undeveloped' decorator was introduced in Table 1:2-1. This combined decorator can be applied to any GSN element that the undeveloped decorator can be applied to, and should be applied to the bottom centre of the element. Example of an undeveloped goal, demonstrating the application of the decorator

1:3.4 Pattern Definition

1:3.4.1 A Pattern is not just the collection of GSN symbols. Additionally there should always be a supporting pattern description that defines the underlying intent and

constraints on its use. The format and presentation of the definition is not prescribed by this standard. A pattern catalogue may be created to collate a series of patterns; where such a catalogue is created the structure and format of the definition should be consistent and each pattern's definition should have a unique {pattern identifier}. The following topics should be addressed in the pattern definition:

1:3.4.1 Name

1:3.4.2 The pattern's name is the label by which the pattern can be identified and should meaningfully communicate the principle argument being presented. It may be accompanied by one or more aliases, which are an alternative identifiers by which the pattern may also be referred to.

1:3.4.2 Intent

1:3.4.3 The intent statement should state clearly what the pattern aims to achieve.

1:3.4.3 Motivation

1:3.4.4 The motivation statement can be used to state why the pattern was created. It could be expressed in terms of previous experiences e.g. as the abstraction of a successfully presented argument, or challenges addressed e.g. argument topics that are often incompletely or poorly addressed.

1:3.4.4 Structure

1:3.4.5 The structure uses the structural and element abstraction notations to present the pattern, clearly indicating where the argument needs to be further developed or populated with details to instantiate the pattern for a specific case.

1:3.4.5 Participants

1:3.4.6 The participants section augments the structure by providing a description of each element. This can provide more complete descriptions, clarify the role of the element in the overall argument and emphasise the aspects that require development or instantiation.

1:3.4.6 Collaboration

1:3.4.7 The collaboration section should describe how elements of the pattern work together to achieve the desired effect, particularly where there are links that are not readily apparent from the argument structure.

1:3.4.7 Applicability

1:3.4.8 The applicability section should state under what circumstances the pattern can be applied, making clear the assumptions and principles underlying the pattern to avoid inappropriate application in a mismatched context. This section should record what contextual information is required in order to apply the pattern.

1:3.4.8 Consequences

1:3.4.9 The consequences section should make clear what work remains after the pattern has been applied. This should highlight where further support to the argument is required, and assumptions that need to be discharged.

1:3.4.9 Implementation

1:3.4.10 The implementation section should communicate how the application of the pattern is carried out e.g. the order in which elements should be developed; communicate hints or techniques that may ease successful application; highlight common or recognised pitfalls with the application of the pattern; and record potential misinterpretation of the terms or concepts in the pattern.

1:3.4.10 Examples

1:3.4.11 It may be useful to provide example illustrations of the application of the pattern, particularly for more abstract patterns. Illustrations should include a typical case and can be supplemented with atypical cases where more than one example is provided.

1:3.4.11 Known uses

1:3.4.12 It may be useful to provide references to known applications of the pattern. These can serve as additional examples.

1:3.4.12 Related patterns

1:3.4.13 This section can be used to reference patterns that are related e.g. addressing the same intent in a different context.

1:3.5 Argument Templates

1:3.5.1 By exception, as an alternative to the obligation to instantiate all patterns elements a completed argument, instantiation can be by means a ‘template argument’ together with instantiation data. This can avoid producing multiple pages of GSN

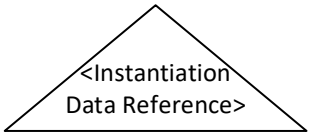
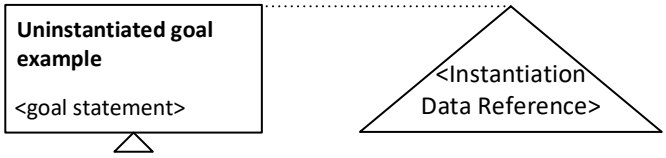
structure where the argument structure is highly repetitive when repeated over multiple aspects e.g. where an argument over requirement satisfaction for individual requirements appeals to different test cases, but otherwise is an identical argument. Table 1:3-3 identifies an additional symbol used to indicate that the GSN argument is a template argument to be instantiated from instantiation data.

1:3.5.2 A template argument is a special case of a pattern argument. It uses the core GSN and argument pattern extension to construct an argument structure which requires no further development. The use of the 'undeveloped' decorator is not permitted within a template argument in its published form. The instantiation data must cover all instantiable aspects including optionality, multiplicity and choice.

1:3.5.3 Where a template argument ends at an element other than a solution, that final element must exist elsewhere within the argument, or in the case of an away element, must be declared in instantiated form in the module interface.

1:3.5.4 Where a template argument and instantiation data is used, it must be possible to apply the instantiation, creating all instantiated versions and meet all the core GSN rules, including uniqueness of element identifiers. Uninstantiated identifiers in a template only need to be unique within the template.

Table 1:3-3 GSN Extension for Templates

GSN Symbol Rendering	Definition
	**Instantiation Data Reference**. This symbol indicates that the GSN argument below the attached element is to be instantiated as a template argument. It provides a reference to the information used to instantiate the template argument. The symbol is not considered a GSN element as it does not form part of the argument. It is attached to the top element of the template argument by a dotted line between the top edges of that element and the symbol (as shown below).
	Example use of an instantiation data reference.

1:4 Modular Extension

1:4.1 Introductory

1:4.1.1 Goal structures can be partitioned into separate, but interrelated, modules. This can allow the division of an overall goal structure into separate goal structures focusing on particular aspects of the overall argument. This section describes how GSN has been extended to represent modular arguments.

1:4.1.2 The concepts of ‘argument view’, ‘architecture view’, module interfaces and inter-module contracts are introduced. It is noted that the concept of an argument module exists even in core GSN, however it is often implicit in non-modular representations.

1:4.1.3 A module may contain one or more arguments and may contain other modules.

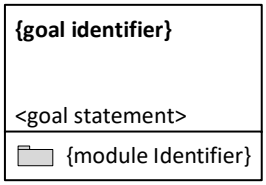
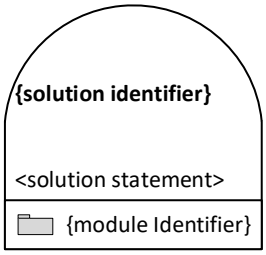
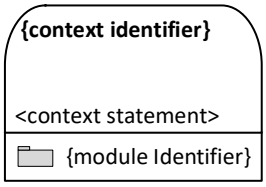
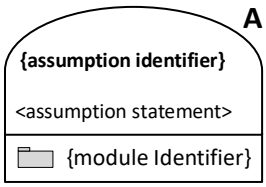
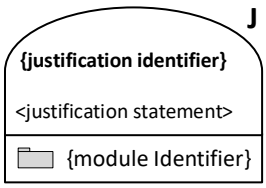
1:4.2 Argument View

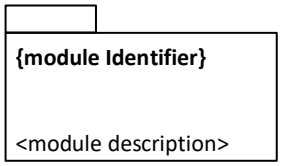
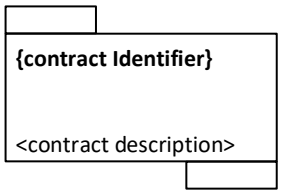
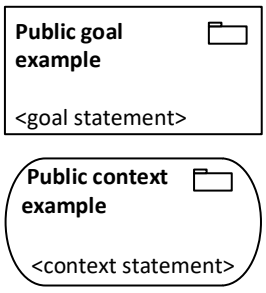
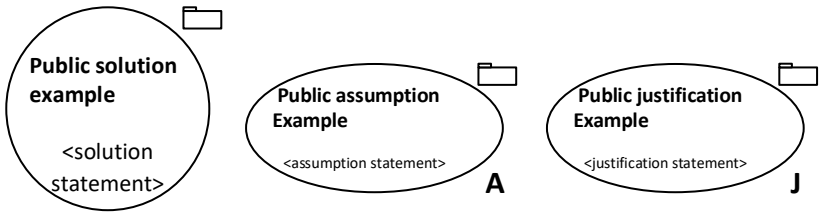
1:4.2.1 The argument view depicts the argument inside an individual module. The following elements are used in addition to the core GSN notation:

- Away Goal
- Away Solution
- Away Context
- Away Assumption
- Away Justification.
- Module Reference
- Contract Reference

1:4.2.2 Table 1:4-1 provide the definition and rendering of these elements whilst Table 1:4-2 provides the extended definition of the existing relationships. The meanings of structures combining these elements are further explained in Section 1:4.3. Note that each argument module has its own namespace for identifiers, thus two elements with the same element identifier can exist in different argument modules. Element identifiers must be unique within a single argument module.

Table 1:4-1 GSN Element Extensions for the Argument View

GSN Element Rendering	Definition
	An **away goal** reference is rendered as a rectangle with a bisecting line in the lower half of the rectangle. The area in the lower portion contains a miniature shaded module element symbol. An away goal reference repeats a claim presented in another argument module.
	An **away solution**, rendered as a semi-circle sitting on top of a rectangle (the semi-circle may be raised above the rectangle by extending its vertical extremes in a straight line). An away solution repeats a reference to evidence items presented in another argument module.
	An **away context**, rendered as shown left, repeats a contextual artefact. An away context repeats a reference to context presented in another argument module.
	An **away assumption**, rendered as a semi-ellipse sitting on top of a rectangle with the letter 'A' at the top-right (the semi-ellipse may be raised above the rectangle by extending its vertical extremes in a straight line). An away assumption repeats an assumption presented in another argument module and is typically used only in Contract Modules.
	An **away justification**, rendered as a semi-ellipse sitting on top of a rectangle with the letter 'J' at the top-right (the semi-ellipse may be raised above the rectangle by extending its vertical extremes in a straight line). An away justification repeats a justification presented in another argument module and is typically used only in Contract Modules.
For all away elements defined above, the element has an identifier which is the {element identifier} of the referenced element in the module in which it was originally declared. The <element statement> contains an exact repetition of the text of the referenced element. The {module identifier} is the identifier of the module in which the referenced element occurs.	

GSN Element Rendering	Definition
	A **module reference**, rendered as a rectangle with a second smaller rectangle adjoining at the top left, presents a reference to a module containing an argument. Note that a module reference points to the totality of the argument contained in the referenced argument module, rather than just to an individual claim. A module reference may be used in support and/or as context for an argument. A module reference cannot be used within a contract module.
	A **contract reference**, rendered as a rectangle with a two smaller rectangles (of equal size to each other) adjoining at the top left and bottom right, presents a reference to a contract module. Note that a contract reference points to the totality of the relationship contained in the referenced contract module, rather than just to an individual claim. A contract reference cannot be used within a contract module.
 Public Decorator Symbol 	**Public Decorator**, rendered as a miniature module symbol and superimposed on a goal, solution, context, assumption or justification symbol at the top right. This indicates that the element is publicly visible in one or more interfaces of the module and can be referenced as an away element. The preferred location of the public decorator is within the element shape. Where this is not practical (e.g. as shown below) the exact positioning of the public decorator is not important as long as the association with the element is clear.
	

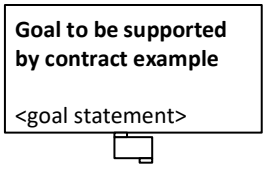
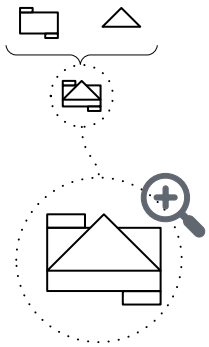
GSN Element Rendering	Definition
 'to be supported by contract' decorator symbol   'to be supported by contract' and 'to be instantiated' decorator	**To be supported by contract:** This decorator, attached centrally immediately below the goal to which it relates, denotes that support for the claim presented by the attached goal is intended to be provided from an argument in another module linked by an as-yet-undisclosed contract. At some later stage, the element may be updated to replace this decorator with support from a named contract module, or may be left as it is, with the necessary support defined in a higher-level argument's architecture view. This decorator can only be applied to goal elements, and can be used in conjunction with the 'to be instantiated' annotation, but is mutually exclusive with the 'to be developed' annotation. This decorator cannot be used within a contract module.

Table 1:4-2 GSN Relationship Extensions for the Argument View

GSN Relationship Rendering	Definition
	**SupportedBy** In addition to the permitted connections defined in the core GSN definition (See Table 1:2-2), in modular GSN the following 'supported by' connections are permitted: goal-to-away_goal, goal-to-away_solution, goal-to-module_reference, goal-to-contract_reference, strategy to away_goal. In a Contract module, the following 'supported by' connections are also permitted: away_goal-to-goal, away_goal-to-strategy, away_goal-to-away_goal.
	**InContextOf** In addition to the permitted connections defined in the core GSN definition (See Table 1:2-2), in modular GSN the following 'in context of' connections are permitted: goal-to-away_goal, goal-to-away_context, goal-to-away_assumption, goal-to-away_justification, goal-to-module_reference, strategy-to-away_goal, strategy-to-away_context, strategy-to-away_assumption, strategy-to-away_justification and strategy-to-module_reference. In a Contract module, the following 'in context of' connections are also permitted: away_goal-to-away_context, away_goal-to-away_assumption, away_goal-to-away_justification.

1:4.3 Argument View Notation Interpretation

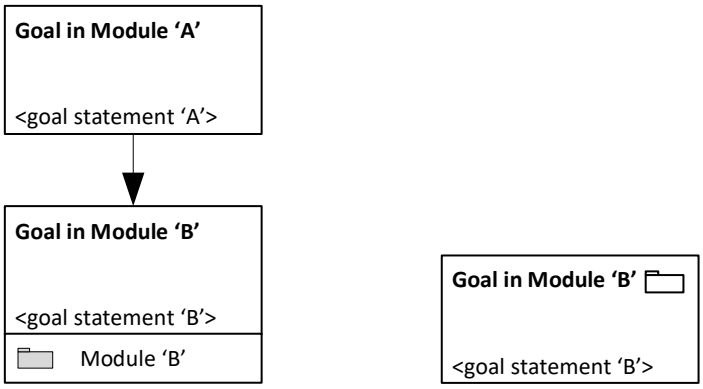
1:4.3.1 The GSN elements defined in Sections 1:2.1 and 1:4.2 above are intended to be combined to represent logical structures. The notation interpretation for core elements within modular extensions is unchanged. Away goals, away solutions and away context elements are used in place of their core counterparts with the addition that they are references to the goal, solution or context in the referenced argument module.

1:4.3.2 Away goals cannot be (hierarchically) decomposed and further supported by sub-elements within the current argument module; rather, decomposition needs to occur within the referenced argument module. By exception, it is valid to decompose away goals within safety case contract modules where they refer to a goal requiring

support from a contract module. Conversely, the goal requiring support, which is addressed via a contract, must not be decomposed in its host module.

1:4.3.3 Arguments supported by another argument module can be indicated in a number of ways. Figure 1:4-1 illustrates a firm relationship by which the parent goal is supported by a specific goal in the referenced argument module. As with core GSN, an intermediate strategy could be shown and the parent goal/strategy could be supported by one or more argument elements in addition to the away goal.

1:4.3.4 By making the relationship to the away goal the author is asserting not only the inference of support for the parent goal, but also that the context in which the away goal is declared is consistent with the context and assumptions in scope for the parent goal.



(a) Module A (b) Module B

Figure 1:4-1 Use of 'Away Goals'

1:4.3.5 Figure 1:4-2 illustrates a relationship where the parent goal is supported by an argument in an unspecified module, where that contract of support relationship is explicitly instantiated within a specified contract module.

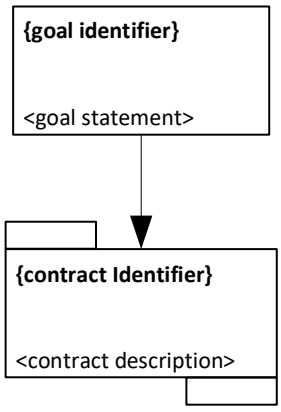


Figure 1:4-2 Use of Contract

1:4.3.6 An alternative approach is illustrated in Figure 1:4-3. The contract module instantiating the support relationship is not specified. Here, the relevant higher-level argument abstraction (e.g. architecture view) should be referred to, which will indicate where the required contract details are specified.

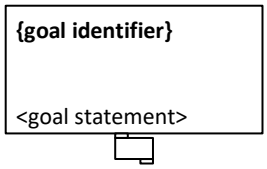


Figure 1:4-3 Use of Unspecified Contract

1:4.3.7 Where a module reference element is shown in support of a parent goal as illustrated in Figure 1:4-4 below, this signifies that the parent goal is supported by the entire argument made in the referenced argument module.

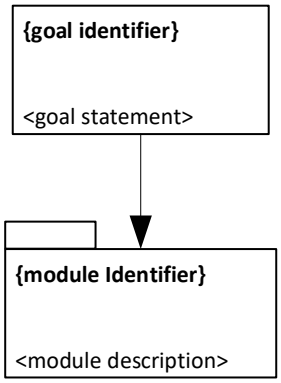


Figure 1:4-4 Use of a Module

1:4.3.8 There may be occasions when a goal or strategy requires fuller justification than can be provided within the confines of a GSN justification element (described in Section 1:2.1 above). In such cases, an away goal can be substituted for the justification. This enables the author to invoke the argument supporting the away goal in the remote argument module as context for the goal or strategy they are currently working with. Use of away goals to replace justification for GSN goals and strategies is illustrated in Figure 1:4-5:

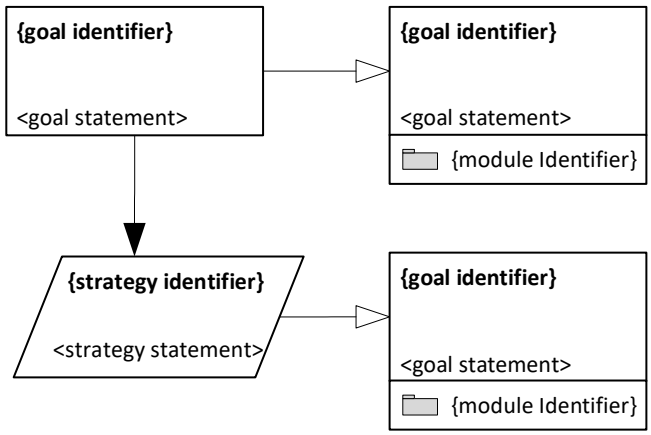


Figure 1:4-5 Use of 'Away Goals' to replace Justification

1:4.4 Architecture View

1:4.4.1 The architecture view provides an abstract view of the relationship between argument modules. The use of links in the architecture view is extended and there is a clear distinction between the use of SupportedBy and InContextOf relationships between individual elements within modules and their use in the architecture view. This is clarified in Table 1:4-4 below.

Table 1:4-3 GSN Extensions to support the Architecture View

GSN Element Rendering	Definition
{module Identifier} <module description>	**Module** symbols are used in the architecture view to represent an argument module. The module identifier may be located internal to the symbol (as shown) or immediately below the symbol. Inclusion of the module description is optional
{contract Identifier} <contract description> {contract Identifier}	**Contract** symbols are used in the architecture view to represent a special type of module that defines the relationship between argument module interfaces and shows how one module supports the argument in another. Alternative contract module symbols are available to suit different styles of presentation of the architecture. The contract identifier may be located internal to the symbol (as shown) or immediately below the symbol. Where the simple form symbol is used the identifier may be located to the side of the symbol. Inclusion of the contract description is optional.

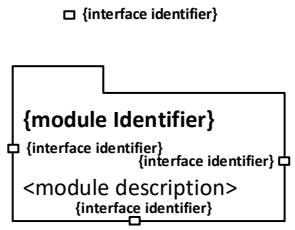
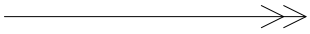
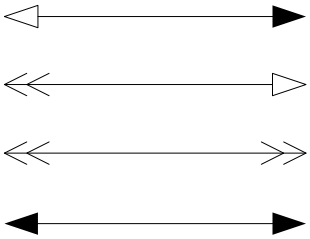
GSN Element Rendering	Definition
	A **Module Interface Connector**, rendered as a small square on the boundary of a module symbol, can optionally be added to aid clarity of the specific interface (specified by the {interface identifier}) used by the inter-module relationship. Where no interface is declared the default interface is assumed. See section 1:4.6 for further details of module interfaces.

Table 1:4-4 GSN Extensions to support relationships between Modules in the Architecture View

GSN Relationship Rendering	Definition
	The **ModuleSupportedBy** and **ModuleInContextOf** relationships are used in the architecture view represent one or more support/context relationship(s) between the elements within the modules. Note that the use of these symbols in the architecture View differs from that within the argument view. In the architecture view the asserted relationship is between modules and may reflect multiple individual support/context relationships across the modules' interfaces.
	The **Composite** Relationship is used where both a supported and a context relationship exists between modules.
	The support/context relationships between modules may be bidirectional, and therefore the relationship may be shown with any of the support, context or composite arrow at either end and in any combination. A small selection of the possible combinations are illustrated (4 out of the possible 9 in addition to the 3 single ended variants)

1:4.5 Architecture View Notation Interpretation

1:4.5.1 It is useful to represent the abstracted structure of an argument in an architecture view. The process of abstraction hides the detailed structure of the argument. Goals, strategies, solutions and context are not shown in the architecture

view; instead, just the modules and their relationships are depicted. The relationships are summarised such that rather than using separate links for each pairing of elements between the modules, only one link is shown.

1:4.5.2 Figure 1:4-6 shows a SupportedBy relationship between modules. The relationship indicates that there exists one or more goal and/or strategy within module 1 which is supported by one or more goal(s) and/or evidence elements within module 2, and similarly for modules 1 and 3. There is no inference that the supporting argument provided in modules 2 and 3 necessarily supports the same goal in module 1. It is entirely permissible for a module both to provide support, and to be supported by another module, provided that this does not create circularity within the argument established by the composed argument modules.

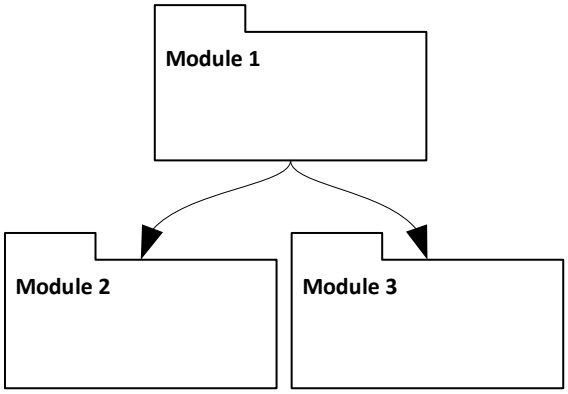
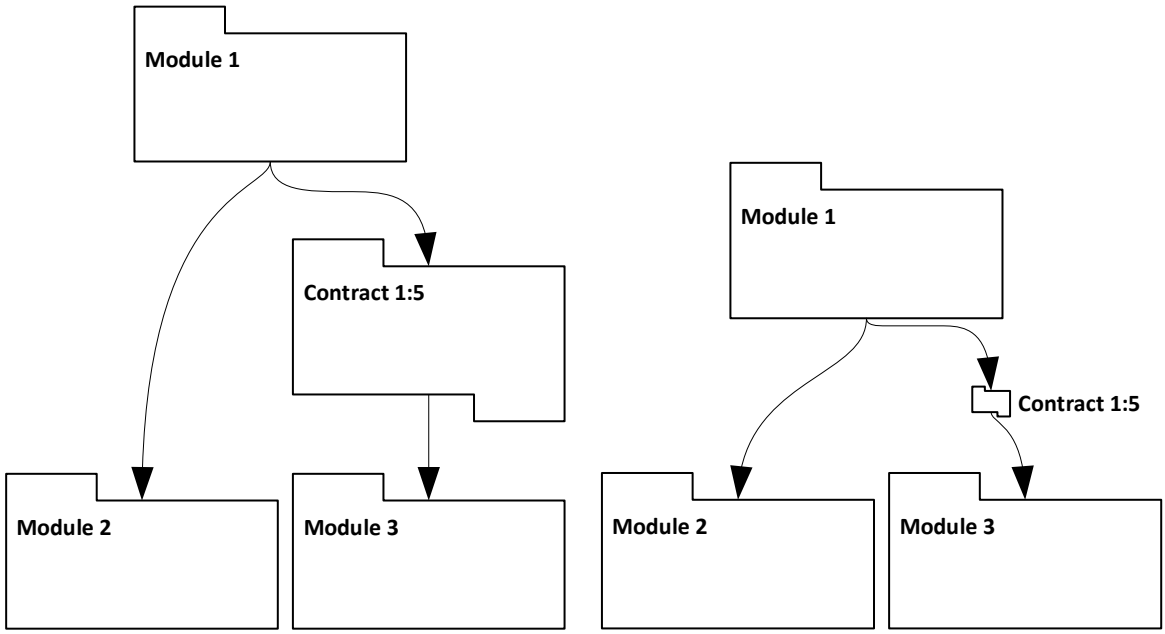


Figure 1:4-6 'Supported By' Relationship between Modules

1:4.5.3 Contract modules can be used in the support relationship between modules to aid decoupling as shown in Figure 1:4-7. Both the full and simple forms of the contract module symbol are shown for comparison. An architecture view may use either form but should be self-consistent. The de-coupling by use of a contract permits argument module construction in cases where the eventual source of support for an argument is unknown at the time of authoring or can be changed for example through re-use or planned product improvement or reconfiguration.



(a) Full form (b) Simple form

Figure 1:4-7 Use of Contract

1:4.5.4 The InContextOf relationship between the two argument modules in Figure 1:4-8 indicates that there exists one or more contextual reference(s) from a strategy/goal within Module 1 to a context element of the argument developed in Module 5:

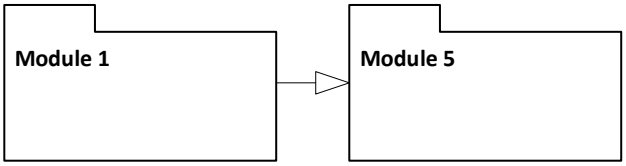


Figure 1:4-8 'In Context of' Relationship between Argument Modules

1:4.5.5 The addition of module interface connectors can aid visualisation of which particular interface is used for a particular inter-module relationship and can be used to give greater clarity where multiple relationships exist between modules. See Section 1:4.6 for further information on Module Interfaces. Figure 1:4-9 indicates that separate interfaces are used for support from module 2 and module 3.

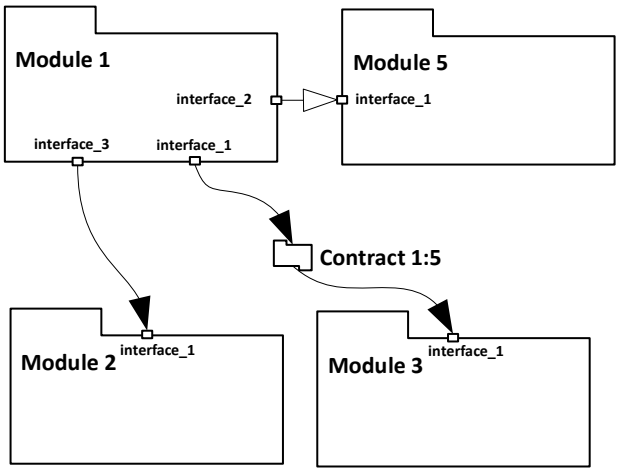
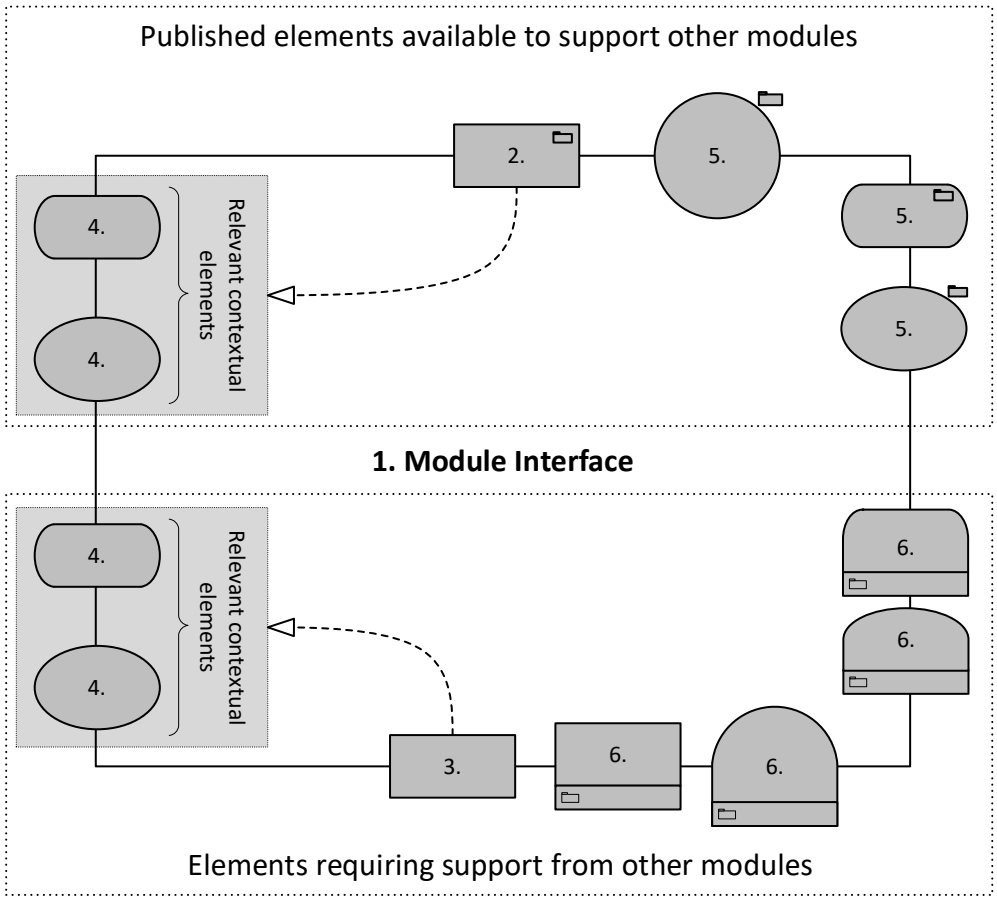


Figure 1:4-9 Use of module interface connectors

1:4.6 Module Interface

1:4.6.1 A Module Interface identifies the published elements of the argument that a module contains. The conceptual content of the module interface is depicted in Figure 1:4-10 and expanded upon in the paragraph that follows.

1:4.6.2 A Module may have one or more interfaces defined, each of which should have a unique {interface identifier}. The default interface publishes all public goals and relevant context together with all goals and references requiring external support. Other interfaces may be published to suit specific purposes and these may be more restrictive than the default interface e.g. to allow detail to be hidden for simplicity or to control exposure of details necessary to relate to peer modules, but unnecessary for integration into a higher level argument.



Note: the numbers in the elements relate to the bullets below

Figure 1:4-10 Module Interface Concept

1:4.6.3 The Module Interface by default contains the following elements; each GSN element should be stated in full including element identifier and the complete element statement:

1. The module and interface identifier, description and configuration information.
2. The goal(s) addressed by the module. These are all the goals declared public using the public decorator within the module. These are not necessarily the ‘top’ goals of a module.
3. Goals requiring support. This should include all those indicated as ‘to be supported by contract’ and any goals requiring support where an explicit dependency has not been declared.
4. The contextual elements (context, assumptions and justifications) relevant to the goals defined above (2 and 3). The interface needs to include all relevant contextual element in scope for that goal, which may be more than the context directly linked to the goal in the argument. Any contextual element included as in scope of a goal in the interface needs to be made public, even if not intended for

reference by another argument module. Contextual elements are specific to each goal.

5. Solutions and context that are available to be cited in support of goals in other argument modules. This includes all solutions and context declared public within the module.
6. Dependencies explicitly referenced within the module. This includes all away-goal, away-solution, away-contextual element references used by the argument within the module. It also includes module(s) and contract module(s) referenced from within the module, together with the goals supported by them.

1:4.6.4 Where a module interface is declared that is a subset of the default interface the sub-set should include all related contextual elements for any goals that are included.

1:4.6.5 Where a module contains other modules, the interface for the containing module can contain any element of the interface of any of the contained modules, in effect promoting the element from the contained module interface to the containing module interface. Where such a promotion occurs, this should ensure that the associated contextual elements for promoted goals are also promoted.

1:4.6.6 The identifiers for all elements within an interface, including that for any promoted element must be unique. Where potential duplication occurs, e.g. where goals of the same identifier are promoted from two contained modules, this can be achieved by including the relevant module identifier, or by introducing an alias for the promoted element.

1:4.6.7 The default interface should maintain full traceability between promoted elements and their originating module, but this does not have to be carried through to an interface that is published for a specific purpose. This abstraction allows an interface to be published without revealing the internal structure of the argument it contains.

1:4.7 Inter-Module Contracts

1:4.7.1 A contract may be used to relate the interfaces of modules to show how the arguments in one module support another. A contract may be described in textual form (e.g. as a table) or for more complex relationships may be described within a contract module using GSN.

1:4.7.2 A contract module is a special type of module that controls the relationship between argument module interfaces using arguments to define how a goal in one module is supported by one or more goals in one or more other modules. It also enables argument to justify the consistency of context between those goals.

1:4.7.3 As the contract module's purpose is to define the relationship between module interfaces it does not have a module interface of its own and cannot publish public elements. All references from the contract module to elements in argument modules must be made using away elements (e.g. away goal, away solution, away context) and can only be made to elements that exist in module interfaces that have been made visible to it.

1:4.7.4 A contract module can contain other modules, however the interfaces for these contained modules are only be available to the contract module in which they are contained, and/or to other modules within the same scope, i.e. they are private to the containing contract module.

1:5 Confidence Argument Extension

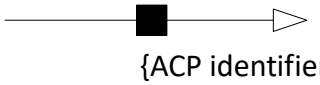
1:5.1 Introductory

1:5.1.1 An Assurance Claim Point (ACP) can be used in GSN to indicate that a confidence argument is associated with an assertion in a risk argument. For each ACP there should exist a corresponding confidence argument.

1:5.2 ACP Notation

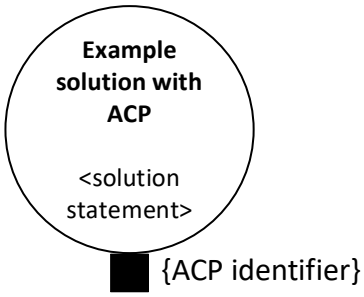
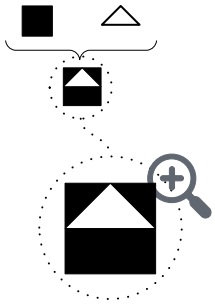
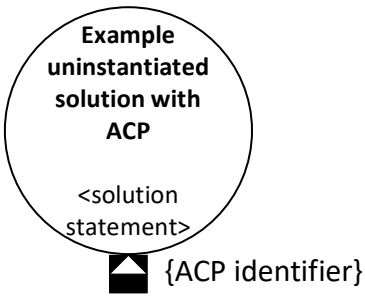
1:5.2.1 Table 1:5-1 illustrates the extensions made to GSN to facilitate the representation of ACPs. These symbols are defined for use as decorators on all core GSN relation types.

Table 1:5-1 ACP Extension (for relationship confidence)

GSN Relationship Rendering	Definition
 	A solid square is the symbol for ACP used as a decorator for a relationship. The label next to the square indicates the ACP identifier. It can be applied to 'SupportedBy' and 'InContextOf' relationships.

1:5.2.2 ACPs may also be added to any element of an argument that provides a reference to an artefact e.g. solution or context where there is a need to argue the confidence in the artefact that the element references rather than the confidence related to its relationship to the argument. Table 1:5-2 illustrates the extension to GSN element by the addition of an ACP decorator.

Table 1:5-2 ACP Extension (for element confidence)

GSN Element Rendering	Definition
{ACP identifier} 	A solid square is the symbol for ACP used as a decorator for an element. The label next to the square indicates the ACP identifier. It can be applied as a decorator to elements that make reference to an artefact (e.g. solution, context).
	The ACP decorator can be combined with the 'uninstantiated' decorator. 

1:5.2.3 Each ACP should have a unique identifier, e.g. "ACP1". The ACP unique identifier should be used to indicate the corresponding argument. The corresponding argument could be located in a paragraph of accompanying text, a goal in the local argument, or a goal in a separate module. Where the corresponding argument is located in a separate module, the module identifier should be shown alongside the ACP identifier delimited with square brackets e.g. ACP1[Confidence].

1:5.3 ACP Notation Interpretation

1:5.3.1 The presence of an ACP indicates that a separate confidence argument documenting the reasons for having confidence in the relationship or referenced

artefact is provided. The nature of confidence arguments is discussed in detail in [8] (Risk, Confidence and Compliance Arguments). The separate confidence argument may be documented in the current argument module, or may be contained in a separate confidence argument module, in which case the ACP identifier is extended to include the {module identifier}.

1:5.3.2 The {ACP identifier} may be a reference to a goal, a section in a document, or other form of unique reference that can be followed by the reader of the argument.

1:5.3.3 In Figure 1:5-1, ACP1 is associated with the inferential relationship between G1 and its supporting goals, G2 and G3, via strategy S1. This relationship is indivisible, such that the confidence argument relates to the entirety of support for G1. The placement of an ACP on an individual ‘SupportedBy’ relationship below the strategy is ambiguous and should be avoided.

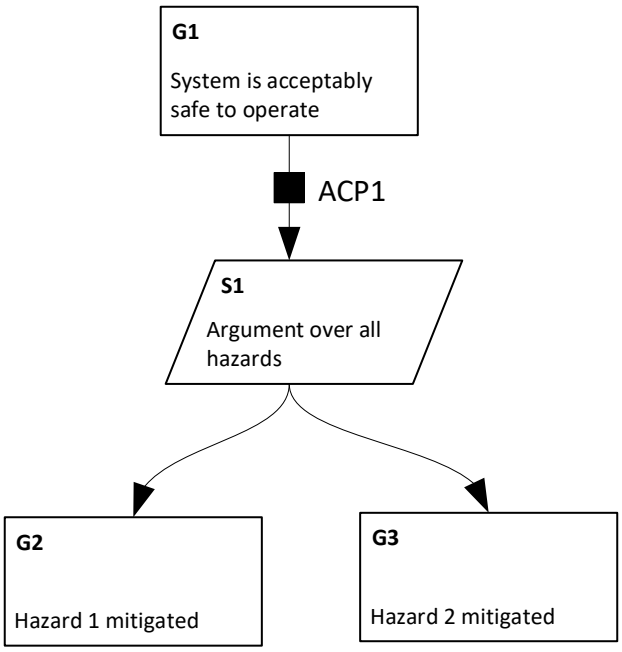


Figure 1:5-1 ACP on Goal ‘SupportedBy’ inference

1:5.3.4 An ACP can be placed on the evidential relationship indicated by the ‘SupportedBy’ relationship between a goal and supporting evidence as illustrated in Figure 1:5-2.

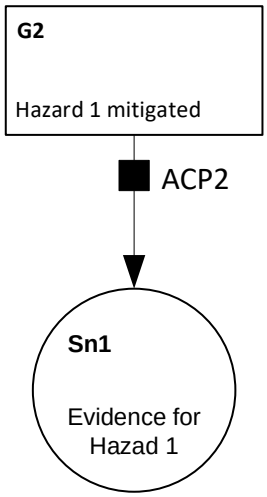


Figure 1:5-2 ACP on evidential 'SupportedBy' relationships

1:5.3.5 Where a single goal is supported by more than one item of evidence, the ACP applies across all 'SupportedBy' relationships in support of the goal and may be illustrated as shown in Figure 1:5-3. This representation may also be applied where a goal is supported by multiple goals without a strategy being explicitly represented.

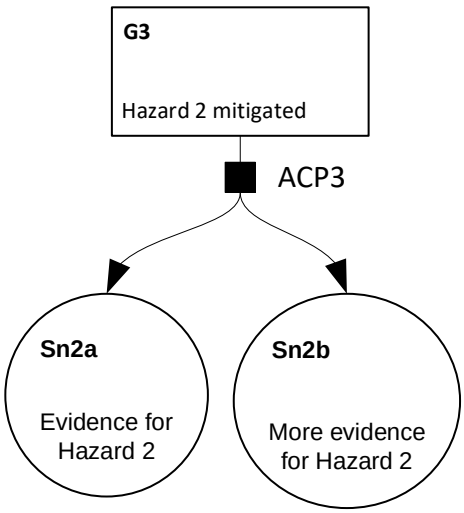


Figure 1:5-3 ACP on multiple evidential 'SupportedBy' relationships

1:5.3.6 An ACP may also be associated with an 'InContextOf' relationship as illustrated in Figure 1:5-4. This enables a confidence argument to support the contextual relationship.

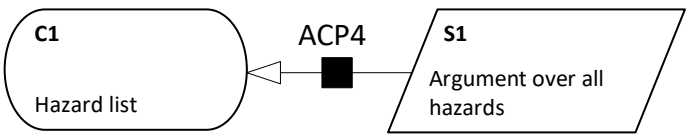


Figure 1:5-4 ACP on 'InContextOf' relationship

1:6 Dialectic Extension

1:6.1 Introductory

1:6.1.1 A Dialectic process in its simplest form is the investigation of truth. Applied to Assurance Cases, dialectics add strength to arguments by comparing options, testing truth, logically disputing and constructively criticising. The use of a dialectic process provides a framework for creating, challenging and questioning Assurance Cases through the discovery and identification of doubt, which can be depicted and the residual doubt exposed.

1:6.1.2 The dialectic extension notation is introduced in section 1:6.2, section 1:6.3 describes the interpretation of permitted combinations of these elements and section 1:2.3 defines the language used within the symbols.

1:6.2 Notation

1:6.2.1 This section defines the notational extension to GSN that support the use of a dialectic process in goal structures.

1:6.2.2 GSN defines dialectic uses of the following core elements:

- Goal
- Solution

1:6.2.3 An additional dialectic specific relationship is provided:

- Challenges

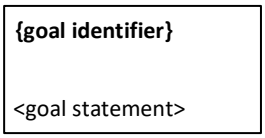
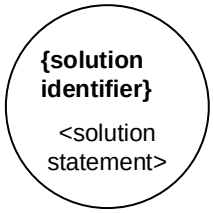
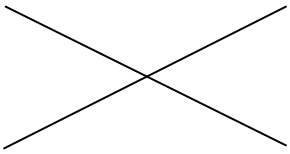
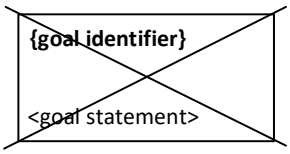
1:6.2.4 GSN defines a status that may be assigned to elements and relationships:

- Defeated

1:6.2.5 Table 1:6-1 provides the definition and rendering of the dialectic elements and GSN relationships are defined in para 1:6.2.7.

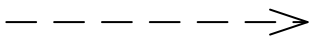
1:6.2.6 The definitions below apply to all the other ‘forms’ of goals and solutions defined within the GSN Extension Tables throughout the standard for the normative definition i.e. Instantiable (represented within curly brackets), instantiated, undeveloped, public/private, away, as applicable.

Table 1:6-1 GSN Element Extensions for Dialectic Arguments

GSN Element Rendering	Definition
	A ***goal***, (core element) can be used in a dialectic context to assert a challenge to part of the argument.
	A ***solution***, (core element) can be used to present a reference to an evidence item that asserts a challenge to part of the argument.
 Defeated Element Decorator Symbol 	***Defeated Element*** decorator symbol, rendered as a cross (‘**X**’) superimposed on a GSN element. This indicates that the element is defeated The *Defeated* decorator can be applied to any of the GSN elements. A ***defeated element,*** *Example:* *(here applied to a Goal)*, rendered with the cross (‘X’) decorator, presents a claim that is defeated.

1:6.2.7 Table 1:6-2 provides the definition and rendering of relationships for use in the dialectic extension. This declares a relationship between a source element (the entity responsible for making the challenge) and a target element. The arrow points to the target. An additional dialectic decorator is also provided.

Table 1:6-2 GSN Relationship Extensions for Dialectic Arguments

GSN Relationship Rendering	Definition
	***Challenges***, rendered as a dashed line with an open arrowhead, allows a Challenge to any GSN entity to be documented. Permitted connections are: goal-to-any element, solution-to-any element, goal-to-any relationship, solution-to-any relationship,

GSN Relationship Rendering	Definition
 Defeated Relationship Decorator Symbol	***Defeated Relationship*** decorator symbol, rendered as a cross ('X') superimposed on a GSN relationship. This indicates that the relationship is *defeated* The *Defeated* decorator can be applied to any of the GSN relationships.
Example of use (SupportedBy relationship) 	A ***defeated relationship,*** *Example:* (here applied to *SupportedBy* relationship), rendered with the cross ('X') decorator, presents a SupportedBy relationship that is defeated.

1:6.3 Notation Interpretation

1:6.3.1 This section introduces the rules which govern the relationships between graphical entities of GSN when extended by the dialectic extension.

1:6.3.2 The dialectic extension can be applied to any existing goal structure that complies with the other applicable normative parts of this standard. These may be in progress or deemed to be complete. Any updates that are required to refactor the structure in order to continue the dialectic process are similarly covered by this standard.

1:6.3.3 A dialectic challenge, can be levied against any part of a goal structure, referred to here as the target of the challenge.

1:6.3.4 A challenge must be levied against the appropriate aspect of the goal structure. For example, it is all too easy to place challenges against a solution (evidence) which is actually valid in its own right, when it is the inference of its use that should be challenged. In such a case, the impact of any resultant defeat on the rest of the goal structure will be unclear and may lead to an invalid goal structure.

1:6.3.5 Counter evidence (via a solution) or an evidenced counter argument (via a goal) can be used to support a challenge to any element in a goal structure e.g. goal, solution, strategy, context, assumption, justification including those that are extended by the other extensions to GSN.

1:6.3.6 Figure 1:6-1 depicts a dialectic challenge to a goal that results in defeat. The dialectic challenge within this structure asserts that if the evidence referred to in Solution CSn1 is valid, this is sufficient to establish that the claim in Goal G1 in the original structure is successfully challenged. Thus, a challenge to a target element is documented by identifying the counter evidence that makes this challenge.

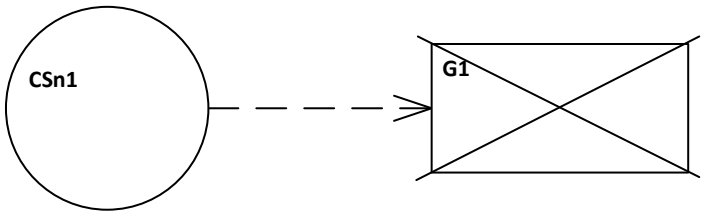


Figure 1:6-1 Goal Defeated by Challenge Through Counter-Evidence

1:6.3.7 In Figure 1:6-1, the challenge made by the evidence presented in solution Sn1 is valid and the claim presented in goal G1 is defeated. The defeat is depicted by the defeated decorator, which is applied to indicate that goal G1 is no longer valid and so presents a claim left as defeated in the goal structure.

1:6.3.8 Figure 1:6-2 depicts a dialectic challenge to a goal. The dialectic challenge within this structure asserts that if the claim presented in Goal CG1 is true then this is sufficient to establish that the claim in Goal G1 in the original structure is in doubt. Thus, a challenge to a target element is documented by identifying a claim that asserts a challenge. The challenge is complete only once an argument to support the assertion is developed and evidenced and so a counter argument is formed.

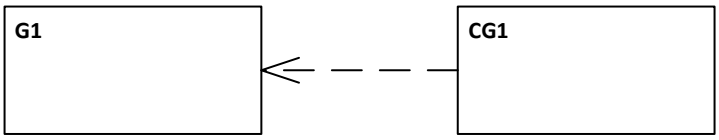


Figure 1:6-2 Goal Challenged Through Counter-Argument

1:6.3.9 Counter evidence (via a solution) or an evidenced counter argument (via a goal) can be used to challenge any relationship in a goal structure i.e. SupportedBy, InContextOf, Challenges, including those that are extended by the other extensions to GSN.

1:6.3.10 Figure 1:6-3 depicts a dialectic challenge to a SupportedBy relationship that results in defeat. This is documented by identifying the evidence referred to by Solution

CSn1 that asserts this challenge. Thus, a successful challenge to a target relationship is developed by applying counter evidence, similarly to Section 1:6.3.6.

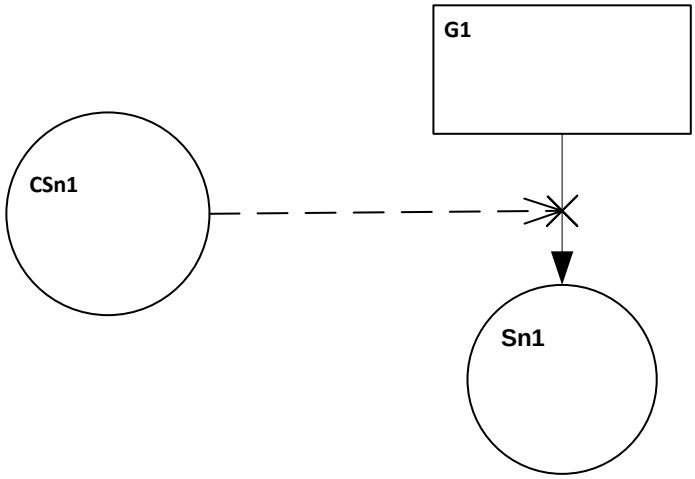


Figure 1:6-3 SupportedBy Relationship Defeated by Challenge Through Counter-Evidence

1:6.3.11 In Figure 1:6-3, the evidence presented in solution CSn1 defeats the SupportedBy relationship. The defeat is depicted by the defeated decorator, which is applied to indicate that the SupportedBy relationship is no longer valid and so is presented as defeated in the goal structure.

1:6.3.12 As the inference between a goal and its supporting goal is indivisible, it is only possible to challenge the inference relationship in its entirety. A challenge cannot be made directly to multiple SupportedBy relationships, so challenges to this inference require a strategy to be inserted. Figure 1:6-4 depicts a dialectic challenge to a multiple SupportedBy relationship that results in defeat. If in the left-hand goal structure (a) the supporting-goals G2 and G3 are considered not sufficient and suitable to support goal G1 and a challenge to this inference is achieved by inserting strategy S1 below goal G1 and applying the challenge to the new strategy, as in the right-hand goal structure (b). The defeat is depicted by the defeated decorator, which is applied to indicate that the strategy S1 is no longer valid.

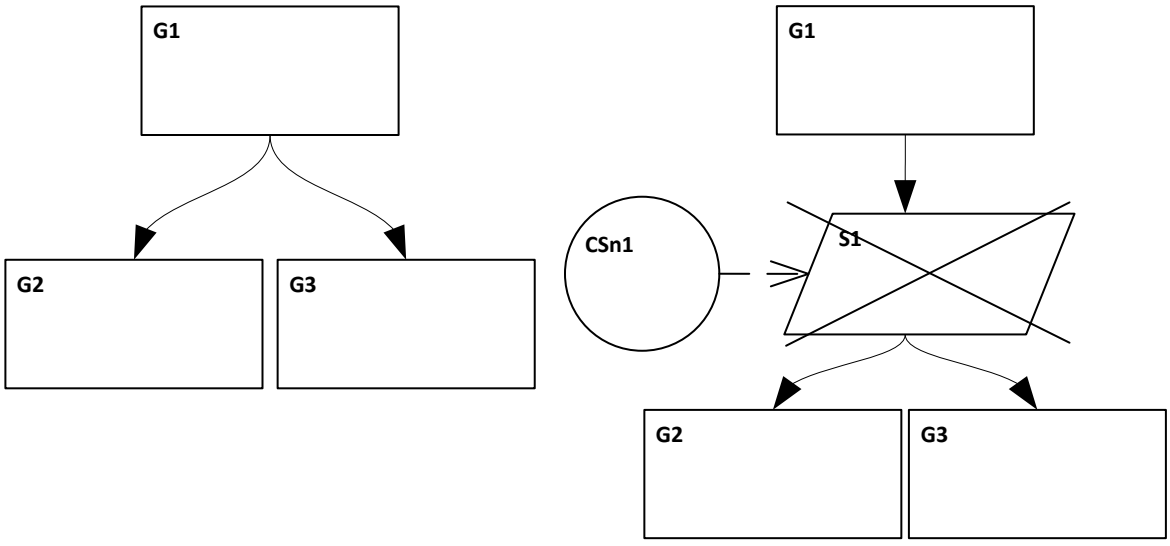


Figure 1:6-4 Multiple SupportedBy Relationships Defeated by Challenge Through Counter-Evidence

1:6.3.13 Figure 1:6-5 depicts a challenge to a Challenges relationship that is documented by identifying a claim represented by goal CG2 that asserts a challenge. Thus, the original challenge can itself be challenged by forming an evidenced counter argument (similarly to Section 1:6.3.8 above). The doubt raised has yet to be resolved.

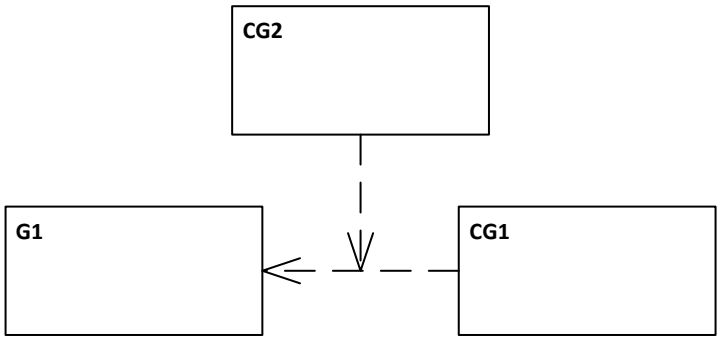


Figure 1:6-5 Challenges Relationship Challenged Through Counter-Argument

1:6.3.14 A challenge may be countered and so may itself be subject to further challenge. A countering challenge may be made to a preceding challenge by challenging: the inference of the challenge (via the Challenges relationship) as in Figure 1:6-5; counter evidence (via the associated solution); a counter claim (via the goal); or any part of a supporting evidenced counter argument that supports the counter claim.

1:6.4 The Language of Dialectics in Goal Structures

1:6.4.1 This section presents a series of simple rules which govern the grammatical structure of statements used in GSN elements when applying the dialectic extension.

1:6.4.2 In a dialectic context, the goal statement is expressed to make a claim that asserts a challenge to a part of the argument.

1:6.4.3 In a dialectic context, a solution references evidence that challenges part of the argument.

1:6.4.4 The goal and solution statements should be clearly expressed such that the crux of the challenge is unequivocally communicated. Thus, the link between the part of the argument that is being challenged (target) and the dialectic element (source) is self-evident.

Part 2: GUIDANCE ON THE DEVELOPMENT AND EVALUATION OF GOAL STRUCTURES

2:1 Introductory

2:1.1 In documenting an argument, an author should address the following objectives:

- The **clarity** of the documented argument – individual claims and references must be easily understandable, and the logical flow of the argument must be clear.
- The **comprehensibility** of the documented argument – author and reader must share an understanding of the claims being made. Where necessary, the author should provide details of the context in which the argument is being put forward and rationale for the argument approach they have adopted and its appropriateness in this context.
- The **veracity** of the documented argument – the documented argument should accurately reflect the true state of the evidence and reasoning at the time of writing.

2:1.2 This part of the Standard is intended to provide pragmatic guidance for authors, to help them produce clear, intelligible and defensible argument structures using GSN. Section 2:2 provides guidance on the layout of GSN goal structures to enable the reader to recognise the logical flow of the argument being presented, and to enhance its readability. Although the development of goal structures is commonly addressed 'top-down', in terms of the decomposition of claims into sub-claims, it is important to note that arguments represented in GSN can actually be developed in other ways including bottom-up or combinations of top-down and bottom-up.

2:1.3 The variety of approaches is reflected in the guidance given in this part of the Standard: Section 2:3 describes top-down approaches to argument development, while Section 2:4 looks at bottom-up approaches. Sections 2:5 and 2:6 address common problems seen in GSN arguments, from the linguistic and structural perspective respectively. Section 2:7 presents a step-by-step process for the review of assurance arguments using GSN.

2:1.4 Version 3 of this standard introduces several new concepts in Part 1. Whilst guidance on each of these new concepts is provided in Sections 2:8 to 2:11, it is acknowledged that the interaction of these new concepts between themselves and with guidance provided for the earlier versions of this standard is not clearly addressed. It is intended that this will be addressed in updates at the next version of the standard.

2:1.5 The examples used in this guidance are typically used in a safety context. This should not be taken to undermine the use of the notation in other contexts.

2:2 Guidance on the Layout of Goal Structures

2:2.1 This section presents brief guidance on the arrangement of GSN elements in goal structures, to enable the reader to perceive the logical flow of the argument being presented, and to enhance its readability.

2:2.2 GSN goals carry the logical burden of the argument, the reasoning that leads the readers to a position where they are able to form a judgement as to whether the argument's conclusion is acceptable. As may be inferred from the language used throughout this Standard to describe relationships between claims, the claims made in GSN goals are stated at different levels of detail. The claim made in the top-level goal (the conclusion of the argument) is stated at a fairly abstract level, and is gradually refined through a series of ever more detailed claims until a direct appeal to some item of evidence is made.

2:2.3 By convention, the claim structure of the argument progresses downwards, from the most abstract claim, recorded in the top-level goal, to an assertion about some item of evidence, recorded in the lowest goal in the structure. The evidence supports the detailed claim immediately above it. The structure is closed out by a reference to the evidence item, recorded in a GSN solution. A GSN structure should be a directed acyclic graph, meaning loops are not allowed.

2:2.4 GSN strategy elements are inserted as required into this vertical claim structure, to provide explanations of the refinement steps between claims made at adjacent levels. Strategy elements should be used as often as is necessary to ensure the inference in goal support is clear.

2:2.5 As discussed in Section 2:3.7, different claims made at the same level of detail may require differing amounts of refinement until they can be closed out. There is no 'right number' of refinement steps. Nor is it advisable to extend the GSN connectors between goal elements to ensure that sibling claims requiring different amounts of refinement should be closed out at the same level on the page.

2:2.6 The conventional layout of a goal structure is that parent goals are located above their supporting goals, and that goals are located above the solutions connected to these goals. Elements connected to goals and strategies using an InContextOf relationship are conventionally laid out to the left and right of those elements.

2:2.7 GSN SupportedBy arrows should emerge from the bottom-middle of the higher-level goals and strategies from which they originate and should connect as closely to the top-middle of the lower-level elements in the relationship as possible.

2:2.8 GSN InContextOf arrows should emerge from the middle of either the left or the right side of the elements from which they originate, and should make the shortest possible connection to the left or right side of the elements to which they connect.

2:2.9 The nature of the arcs describing InContextOf and SupportedBy relationships – in terms of whether the lines are straight or curved, or have bends or corners in them – makes no difference to the semantics of the relationship they assert.

2:2.10 Where goal structures extend over several pages, it is usual to provide an off-diagram connector to allow readers to navigate between pages. The Standard does not mandate any form for this connector.

2:3 Developing Goal Structures Top-Down: The GSN Six-Step Method

2:3.1 This section describes a staged approach to the top-down development of goal structures using GSN. It derives largely from [4]. A running example, representing a partially-developed assurance argument for a fictional automated press system, is used to clarify concepts introduced during the discussion.

2:3.1 Overview

2:3.1.1 Kelly [4] defines six steps in the top-down development of a goal structure:

1. Identify the goals to be supported;

2. Define the basis on which the goals are stated;
3. Identify the strategy used to support the goals;
4. Define the basis on which the strategy is stated;
5. Elaborate the strategy (and proceed to identify new goals – back to step 1), or step 6;
6. Identify the basic solution.

2:3.1.2 Figure 2:3-1 illustrates this six-step process, which is recursive. Having first identified a claim and represented it using a GSN goal (step 1), an explicit statement of the context in which it is valid is made (step 2). A strategy to support it is then identified (step 3) and justified (step 4). In some cases, it may be possible to support the claim immediately through reference to some basic evidence (step 6). More commonly, however, it will be necessary to identify some intermediate sub-claims, to refine the argument, incrementally, to a level of detail at which the claim can be stated at a sufficient level of detail to enable it to be supported by basic evidence (step 5). In such cases, the process begins again at the next level of detail, starting from the newly-identified goals (step 1).

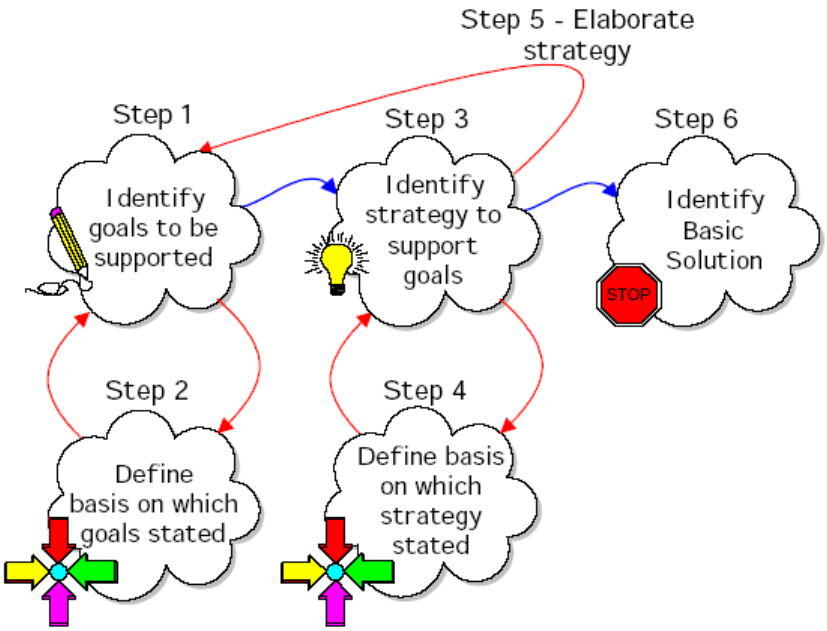


Figure 2:3-1 Six-Step Process for Developing Goal Structure

2:3.2 Step 1: Identify Goals

2:3.2.1 The objective of this step is to identify the top goal(s) of the structure, the principal claim(s) that the remainder of the argument should support. It is important

that the claim made in the top goal is stated at an appropriate level of detail. It is imperative that the author consider the reader's likely response here. If the claim jumps ahead of a more fundamental objective, this risks the reader drawing conclusions at too low a level and precludes the demonstration of the derivation of the top-level claim from that fundamental objective. Figure 2:3-2 introduces the top goal of the running example used to illustrate the top-down development of a goal structure in this section:

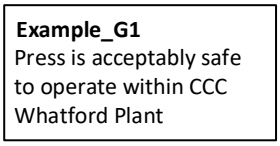


Figure 2:3-2 Top Goal of Running Example

2:3.3 Step 2: Define the Basis on which Goals are Stated

2:3.3.1 A claim made in a goal structure (or, indeed, in any other argument structure) can be evaluated as 'true' or 'valid' only if the basis on which it is stated is clear: no claim can be assumed to have 'universal validity'. It is the author's role to ensure that the reader has an adequate, and correct, understanding of the context surrounding the claim, so that they are able to form a judgement as to how convincing it is. In step 2 of the method, the author constructs an explicit record of the information necessary for the reader to understand the context in which the claims identified in step 1 are put forward. There are three key aspects to this activity:

- Identifying information required about the system under discussion;
- Identifying information required about the context of the system;
- Identifying information required about the argument (for example, definitions of terminology used).

2:3.3.2 GSN contexts are used to refer to system information, artefacts or processes. Figure 2:3-3 illustrates the association of context with a goal, to clarify concepts introduced in the claim:

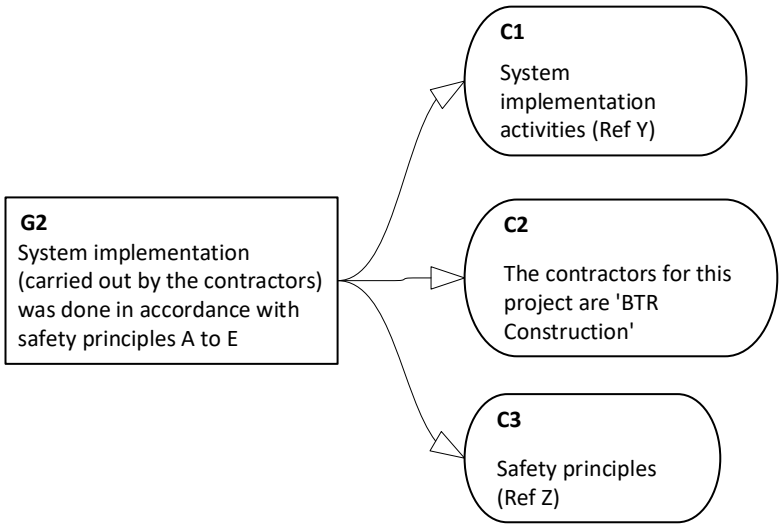


Figure 2:3-3 Association of Additional Contextual Information

2:3.3.3 In Figure 2:3-3, the claim made in Goal G2 introduces three terms which potentially require clarification for the reader: “system implementation”, “the contractors” and “safety principles A to E”. Contexts C1 and C3 refer to the system and process artefacts which clarify the first and third of these concerns. Context C2 provides an explanation of the second.

2:3.3.4 Note that contextual information associated with a claim made in a particular goal is understood to be in scope for all goals which support that goal. Therefore, in determining whether additional context is required, goal-statements should be examined for terms and concepts which have not been defined within the inherited scope. Since a contextual statement makes an assertion in the argument structure, nothing in the supporting argument for the goal to which the context is applied should contradict or undermine the relationship between the goal and the context.

2:3.3.5 It should be noted that it is not always appropriate or necessary to define every term used within a goal-statement. Firstly, the objective of using context is to ensure that there is a clear understanding of goal-statements between reader and writer. In some cases, this can be relied upon without further definition, as for example in the case of terms and concepts which are commonplace and well understood by both parties. Secondly, definitions can be provided throughout the course of the argument communicated by the goal structure. For example, consider the case of a top-level goal “System X is safe”. This statement appears to contain two terms requiring definition: ‘System X’ and ‘safe’. ‘System X’ can be clarified by reference to some model information using a GSN context element. However, it is the purpose of the

goal structure to argue the meaning of the word ‘safe’ - the term ‘safe’ is defined by whatever argument is put forward in support of this top-level goal. Therefore, at the top level in the goal structure, ‘safe’ can legitimately be left without explicit definition.

Example

2:3.3.6 Figure 2:3-4 represents the top goal of the argument which is used as a running example to demonstrate the gradual development of a GSN goal structure from the top down. The argument’s top-level claim is documented in Goal Example_G1:

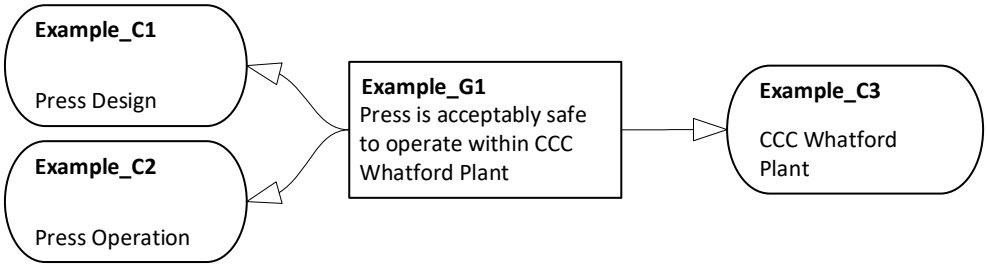


Figure 2:3-4 Example with Contextual Explanation

2:3.3.7 In Figure 2:3-4, the terms “press”, “operate” and “CCC Whatford Plant” have been drawn out into explicit GSN context elements, which provide reference to the artefacts in which they are fully defined. The concept “acceptably safe” is left to be defined through the supporting argument.

2:3.4 Step 3: Identify Strategy

2:3.4.1 Having identified and expressed a claim and explicitly stated the context in which it is stated, the author’s next task is to work out how the claim can be substantiated. Again, a consideration of the reader’s likely reaction is a useful guide. The author should ask himself the following questions:

- What reasons are there for saying that the goal is true?
- What statements would convince the reader that the goal is true?

2:3.4.2 The intention is to find argument approaches (strategies) which will give rise to further goal-statements which are, in some way, easier to support than the overall claim. One such strategy would be a ‘Divide and Conquer’ approach, by which a high-level goal is decomposed into a number of ‘smaller’ goals, the satisfaction of all of which would be sufficient to support the original goal. Figure 2:3-5 illustrates this approach:

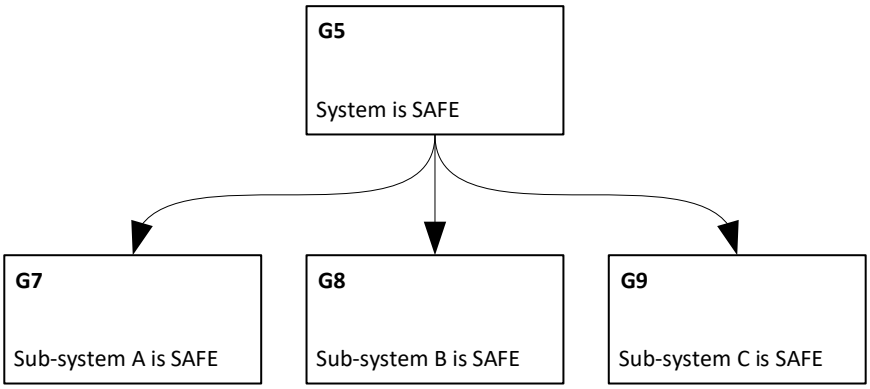


Figure 2:3-5 Divide and Conquer Goal Decomposition

2:3.4.3 Another common approach is to attempt to re-state the original claim as one more closely related to the specific application in question or to the evidence that will ultimately be used to support the argument. Figure 2:3-6 illustrates this approach:

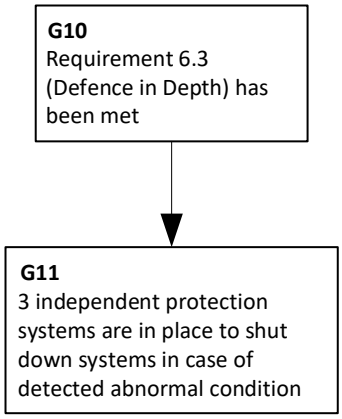


Figure 2:3-6 Interpretation, or Particularisation, of a Goal

2:3.4.4 Argument approaches such as those described above are represented in GSN by the use of strategy nodes. The role of a strategy node is to explain the logic which connects the statement made in a parent goal with those made in the supporting goals derived from it. It can be helpful to think of the role of a GSN strategy as analogous to an explanation included between two lines of working in a mathematical calculation, as follows:

$3xy^3 + 2x^2y^2 + 5xy = 17y$ (Divide both sides by y)

$3xy^2 + 2x^2y + 5x = 17$

The strategy adopted here is to divide both sides of the equation by y. Providing an explicit explanation allows readers to understand the flow of the logic more clearly and

also provides a basis from which it is possible to check that the strategy has been applied correctly.

Example

2:3.4.5 Figure 2:3-7 shows the strategies that have been identified as approaches to arguing that the press is acceptably safe. Strategies S1 and S2 provide an explicit indication of the two 'strands' of argument which are being put forward to support the claim made in Goal G1:

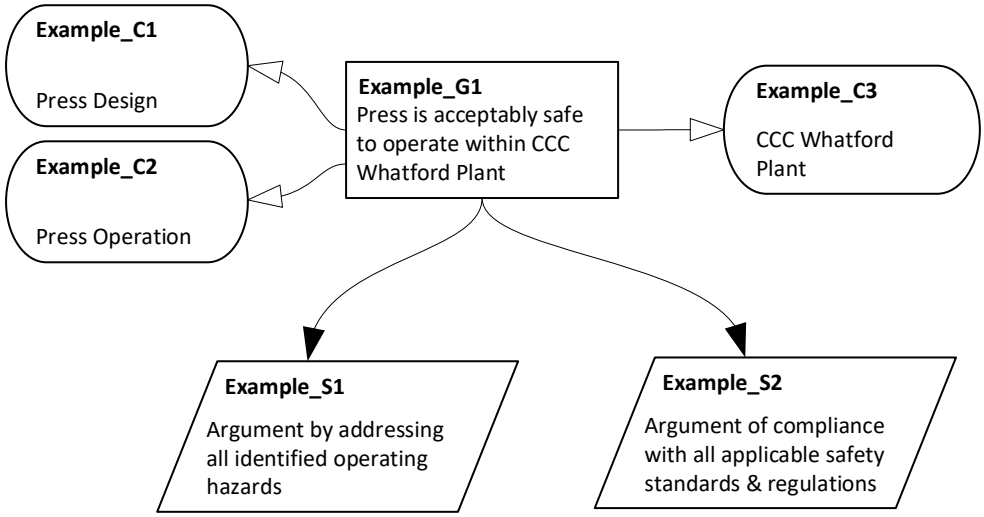


Figure 2:3-7 Example with Top-Level Strategies

2:3.5 Step 4: Define the Basis on which the Strategy is Stated

2:3.5.1 It is necessary to define the basis on which an argument strategy is stated, so that its validity can be assessed, just as, in Step 2, goal-statements required an explicit statement of the context in which they are stated. This involves identifying the contextual information required to understand the argument approach described by the GSN strategy node and to use the strategy to derive goals at the next level of detail. The process of identifying context for strategies is the same as that for goals described in Step 2: strategies should be examined and assessed for terms or concepts that have been introduced but not defined explicitly. For example, the simple system decomposition strategy that was shown in Figure 2:3-7 refers to "all identified operating hazards". Information must be associated with the strategy to define this term for the system in question, so that the decomposition can be carried out properly at the next stage.

2:3.5.2 As well as definitions of terms, the contextual basis for the argument strategy may include rationale information as to why the strategy has been adopted. In GSN, this is achieved with the use of assumptions and justifications. GSN assumptions record any assertions made about the system, its operating context, users or environment that the strategy depends on. Note that an assumption is a proposition that is asserted as true and therefore is not supported by further argument. Justifications record the reasons why a given strategy is proposed as an approach to supporting a particular goal, or provide reasons why the strategy being adopted is adequate.

Example

2:3.5.3 Continuing the development of the goal structure, Figure 2:3-8 shows the contextual information necessary to clarify Strategies S1 and S2:

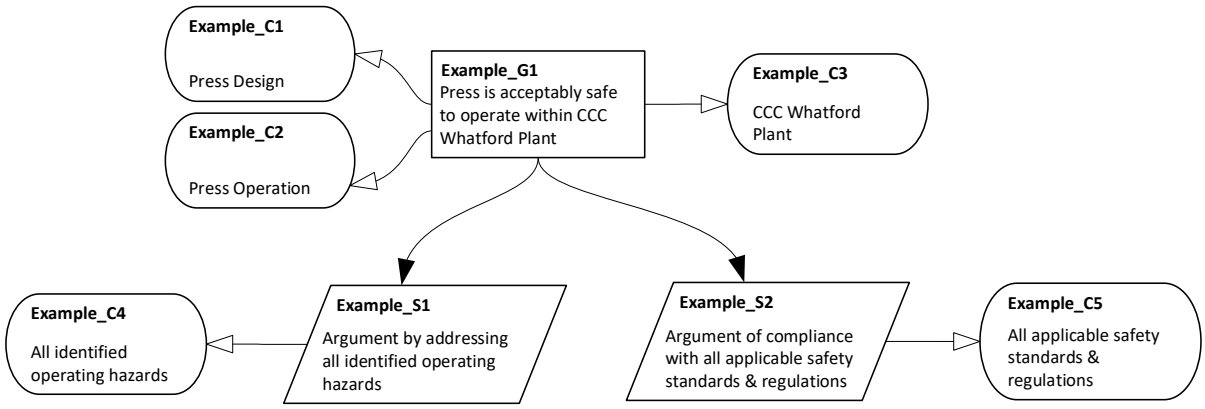


Figure 2:3-8 Example with Contextual Evidence to Clarify Strategies

2:3.5.4 No justification of the strategies has been provided here. If the author feels that the reader might question the suitability or adequacy of the argument approaches adopted, appropriate justifications should be attached to the strategy elements. Similarly, if any significant assumptions were made in determining the argument strategy, these should also be recorded.

2:3.6 Step 5: Elaborate the Strategy

2:3.6.1 Once the argument approach has been decided, it is enacted and the goal-statements that follow from its application are identified. It is important to note that the argument itself is contained in and carried by the structure of claims recorded in goals at different levels of detail: the GSN strategy is merely a means of clarifying how these are related to one another. For example, for a strategy which states that an argument

is made concerning all of a system's constituent sub-systems, appropriate claims are made for each of the defined sub-systems. Similarly, if the strategy states that a quantitative argument approach should be adopted, quantitative claims must now be put forward as goals. Step 5 can thus be thought of as 'putting flesh on the bones' of the strategy identified and clarified in Steps 3 and 4.

2:3.6.2 In some cases, it may be appropriate to leave a strategy implicit, and decompose a goal directly into supporting goals, rather than using an explicit GSN strategy element. It is important to realise that, logically, there is always a strategy underlying the argument's construction.

2:3.6.3 Elaborating a strategy involves defining new goals, i.e. beginning the argument development process again at Step 1, although this time obviously the goals are one level further down the goal structure.

2:3.6.4 It should be noted that a goal stated as part of a strategy in support of a particular parent goal may also form part of the supporting argument of other parent goals.

Example

2:3.6.5 Figure 2:3-9 shows the elaboration of the strategies defined in Figure 2:3-8. Elaboration of strategy S1 involves putting forward an appropriate claim for each of the operating hazards referenced in context C4 (goals G2, G3 and G4). Similarly, the elaboration of Strategy S2 is directed by the list of relevant standards referred to in context C5. Once these have been identified, the argument is developed by putting forward a claim of compliance for each identified standard (goals G5, G6 and G7).

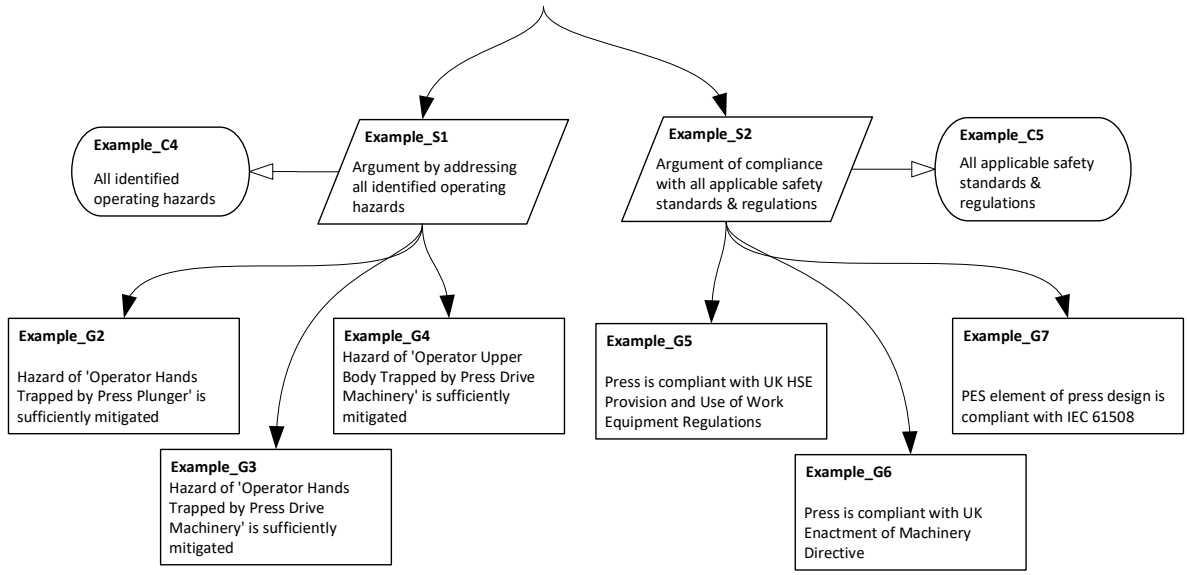


Figure 2:3-9: Elaboration of Strategies

2:3.6.6 The goal structure continues to be developed in this way until it is clear that no further decomposition into supporting goals is necessary and the goal can be directly supported by appeal to some evidence item (Step 6).

2:3.7 Step 6: Identify Solutions

2:3.7.1 Eventually, claims will be expressed at a sufficiently basic level that they do not require further expansion, refinement or explanation, and can be supported by direct reference to eternal evidence. In GSN, a solution element is added to support the goal. The solution provides a reference to some evidence item. Figure 2:3-10 shows the fragment of goal structure developed to support goal G3 in the example, which was derived from the application of strategy S1 in Step 5 (Figure 2:3-9). The claim “Motor/clutch/drive belts surrounded with safety cage” is ‘bottomed out’ with an evidential claim about the adequacy of the press design, supported by an inspection report.

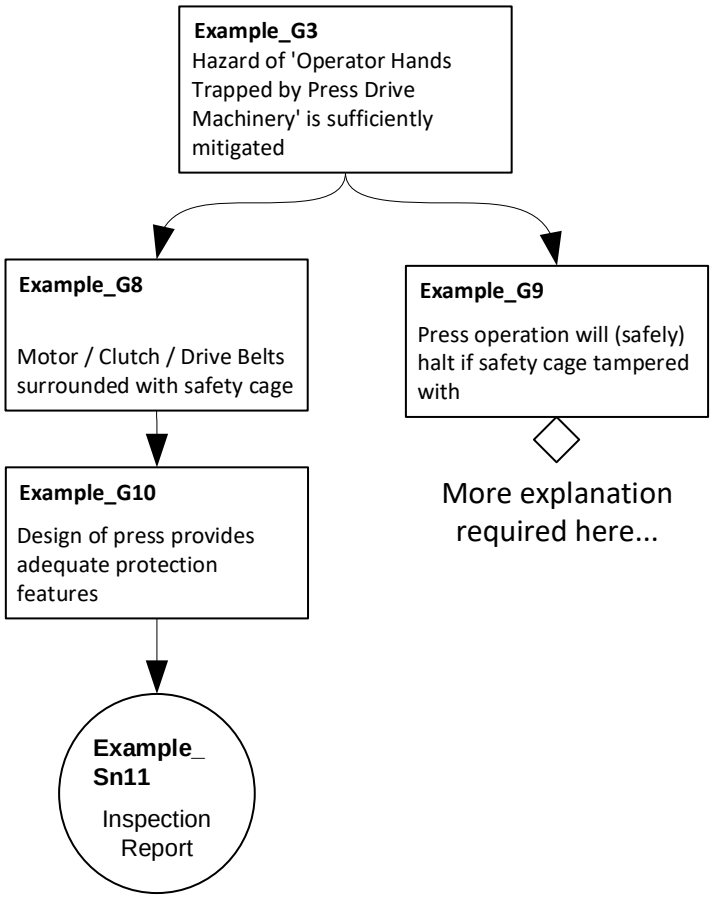


Figure 2:3-10: Reference to Evidential Support

2:3.7.2 Note that peer goals do not always require the same level of decomposition: although Goal G8 is closed out at this level, its sibling Goal G9 requires further argument to bring it to a point at which it can be supported directly by evidence.

2:3.7.3 It is regarded as best practice that the goal most immediately supported by a solution element should be an unambiguous assertion of the property of the evidence item that is being referred to by the argument.

2:3.7.4 GSN solution elements should refer unambiguously and precisely to the section in an evidence item which is required to support the claim in the goal element. References to whole documents should be avoided where possible. However, it is important to ensure that this requirement does not lead to an unnecessary proliferation of evidence assertion claims at the bottom level of the goal structure, i.e. references to large numbers of individual tests when a generic reference to ‘unit test results’ would be adequate.

2:3.7.5 It is possible to cite multiple solutions as providing evidential support for a particular parent goal. However, one drawback of doing this is that the specific contribution each item of evidence makes towards supporting the goal may become unclear. This can be improved through adding an intermediate level of goals, and maintaining a one-to-one association between goals and solutions.

2:3.7.6 It should be noted that a solution stated as providing evidential support for a particular parent goal may also form part of the cited evidential support for other parent goals.

2:3.8 What if the argument can't be closed out?

2:3.8.1 A frequent problem in top-down argument development is that the author gets some way in the decomposition of the claim to be closed out and then realises that the evidence required to 'close out' the claim is missing. Either the required evidence is missing entirely, or, as is more frequently the case, the existing evidence does not 'cover' the lowest-level claim adequately. If a search for additional evidence to provide adequate backing for the claim as it stands is not successful, the argument must be reworked, to take account of the shortcomings. In such circumstances, the author must examine the available evidence carefully (as described in the bottom-up argument development approaches described in Section 2:4 below), and establish the claim that can be made. The claim immediately above the GSN solution element must then be rephrased to accommodate this. This may imply making the claim less specific, or bounding it more carefully. Rephrasing of this kind implies a weakening of the claim made. Having done this, the author must work back up the argument structure, revisiting all of the higher-level claims dependent on this revised claim, to establish whether they are affected by the weakening of the claim. Several higher-level claims may need to be rephrased, at this stage, and the result may be an overall weakening in that strand of argument.

2:4 Developing Goal Structures Bottom-Up: Working from Available Evidence

2:4.1 Introductory

2:4.1.1 It is sometimes necessary or useful to build a GSN argument bottom-up, starting with the evidence available. This might happen, for example, in cases where various analyses, tests etc. have been carried out but where there was originally no intention or requirement to produce a formal assurance case, or in situations where an existing assurance case must be updated or improved. Production of an assurance case, even belatedly, can alleviate the 'evidence without argument' problem inherent in some projects, where collections of safety reports are presented to stakeholders or certification authorities without any coherent explanation as to what they are intended to demonstrate.

2:4.1.2 Adapting Kelly's six steps [4] (see Section 2:3) for top-down GSN development, the following process can be used to develop a goal structure from the bottom up:

1. Identify evidence to present as GSN solutions;
2. Infer 'evidence assertion' claims to be directly supported by these solutions, and present these as GSN goals;
3. Derive higher-level goals that are supported by the evidence assertions;
4. Describe how each layer of goals satisfies the parent goal (i.e. strategy);
5. Check that any necessary contextual information is included;
6. Check back down the structure for completeness;
7. Join the resulting goal structure to a known top goal or a set of goals.

Figure 2:4-1 shows these steps graphically:

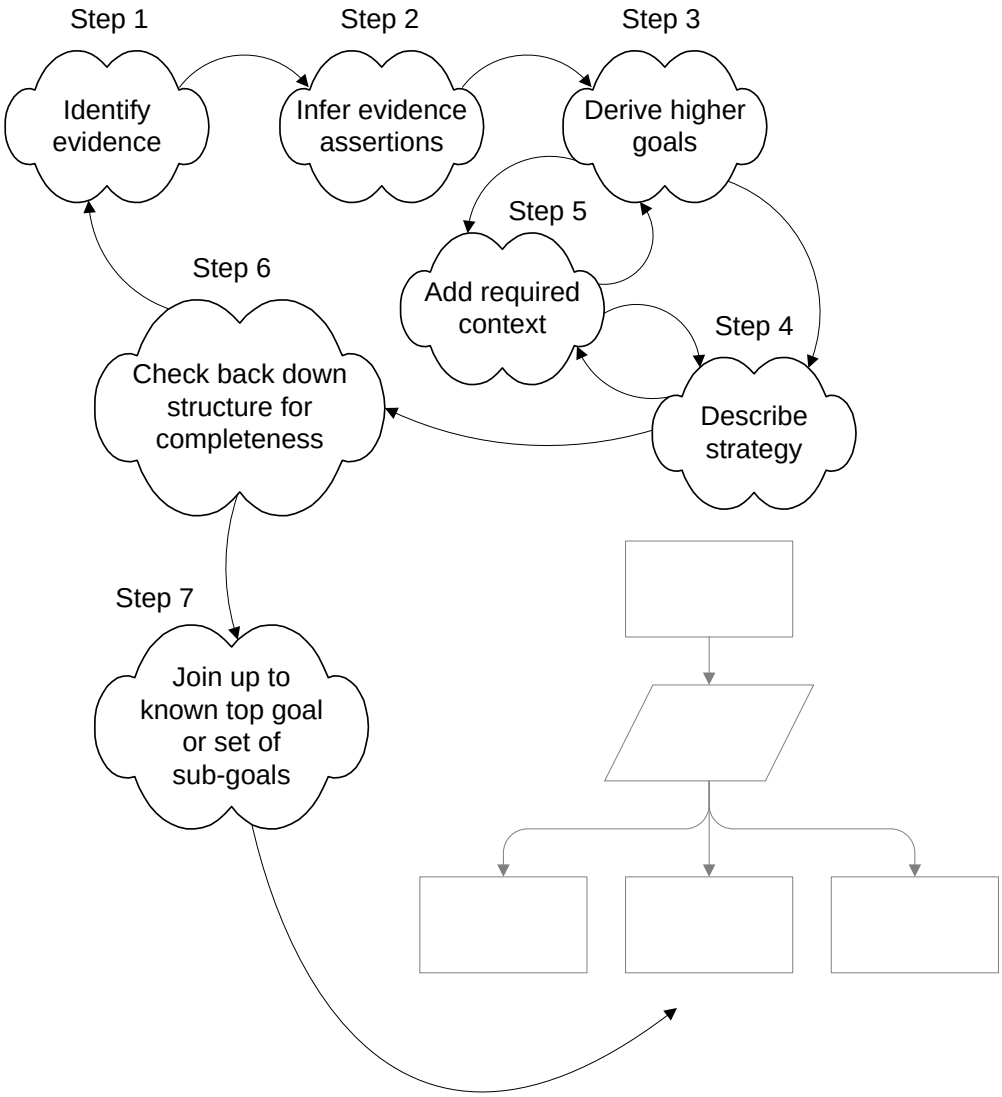


Figure 2:4-1 Bottom-Up Process for Developing Goal Structures

2:4.1.3 During the whole process, the author should keep in mind “what makes the system safe” and write the goal structure to suit. For example, it may be that the safety of a given system depends entirely on physical features e.g. a geographical layout or the provision of interlocks, rather than on its having been developed to a specific process.

2:4.1.4 This approach takes considerable skill and intuition to elicit the appropriate claims from the evidence and ‘spot’ the useful combinations that are likely to converge in support of the desired top goal. It is therefore recommended that this approach is only used by those who are already experienced in developing GSN arguments.

2:4.1.5 The bottom-up approach will rarely be used in isolation to form a complete goal structure. It is more likely that the resulting goal structure will 'join' to a desired higher-level claim that is already understood to be a requirement of the assurance case.

2:4.2 Bottom-Up Step 1: Identify Relevant Evidence

2:4.2.1 In developing a GSN assurance argument bottom-up, the starting point is obviously to ascertain what evidence for system safety exists, and precisely what can be claimed for it. Typical safety evidence would include Fault Tree Analysis (FTA) and Failure Modes Effects and Criticality Analysis (FMECA), shown in Figure 2:4-2:

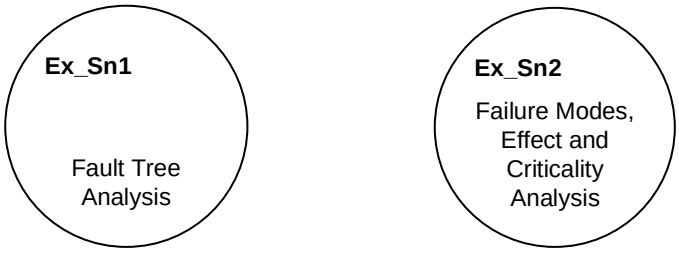


Figure 2:4-2 Typical Solutions Derived from Evidence

2:4.2.2 Having created evidence items from analysis, the author should consider what this evidence reveals about why the analysis was originally carried out. In many cases, this will have been in response to some safety requirement stated in another document, typically a hazard analysis report. This may guide the author towards the types of claims (both quantitative and qualitative) which these evidence items will support (see Section 2:4.3 below).

2:4.3 Bottom-Up Step 2: Infer 'Evidence Assertion' Goals

2:4.3.1 The evidence should be examined carefully, with the question: "What safety claim or property of the system is demonstrated or supported by this item of evidence?" In many cases, the content of the evidence item will suggest a claim, which is represented as a bottom-level 'evidence assertion' goal in the assurance argument (see Section 2:4.7), inferred directly from the available evidence. They differ from higher claim in that the subject is the evidence rather than the system property in question. Figure 2:4-3 demonstrates the inference of evidence assertion goals directly from solutions in GSN:

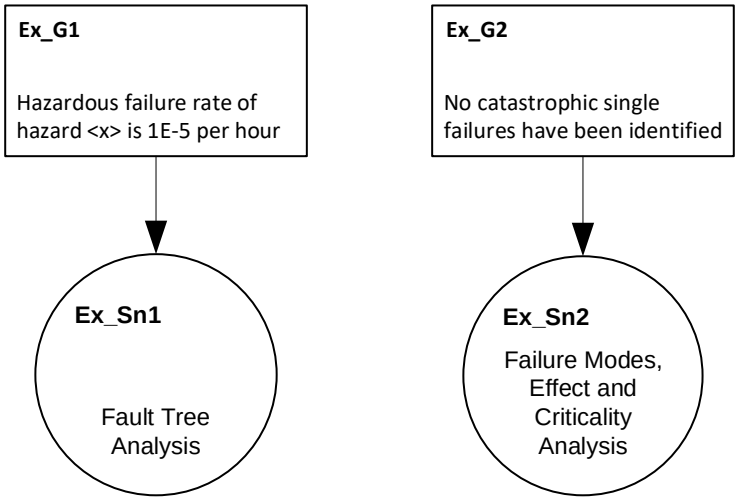


Figure 2:4-3 Evidence Assertion from Goals Inferred from Solutions

2:4.3.2 The goals documented using this approach can then be built into the assurance argument using the process described in Bottom-Up Step 3 below.

2:4.3.3 A given item of evidence may in fact provide support for several goals. If this is the case, the GSN solution attached to each 'evidence assertion' should refer to the individual section of the evidence item which is most relevant to it (e.g. to a paragraph or chapter in a report), if possible. Figure 2:4-4 illustrates this approach:

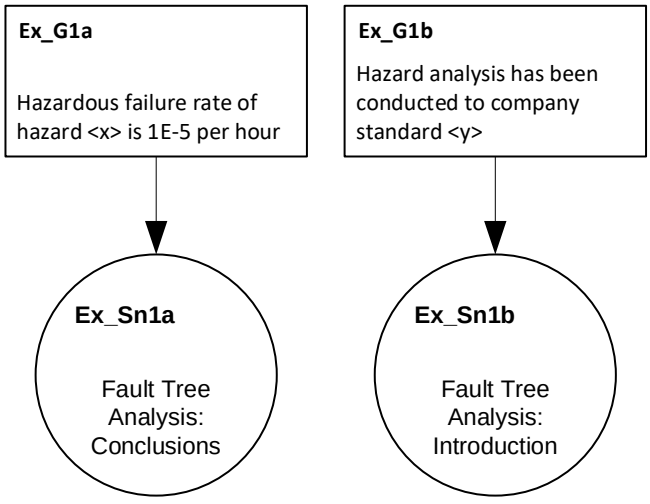


Figure 2:4-4 Multiple Evidence Assertion Goals Inferred from Similar Solutions

2:4.4 Bottom-Up Step 3: Adding Higher Goals

2:4.4.1 Having constructed the bottom of the goal structure as a series of solution elements (representing the available evidence) and evidence assertion goals derived from the solutions, the next step is to work higher in the argument to add a further

hierarchy of goals and strategies. This iterative step is often aiming towards a desired or existing higher-level claim. Figure 2:4-5 illustrates adding a higher-level goal:

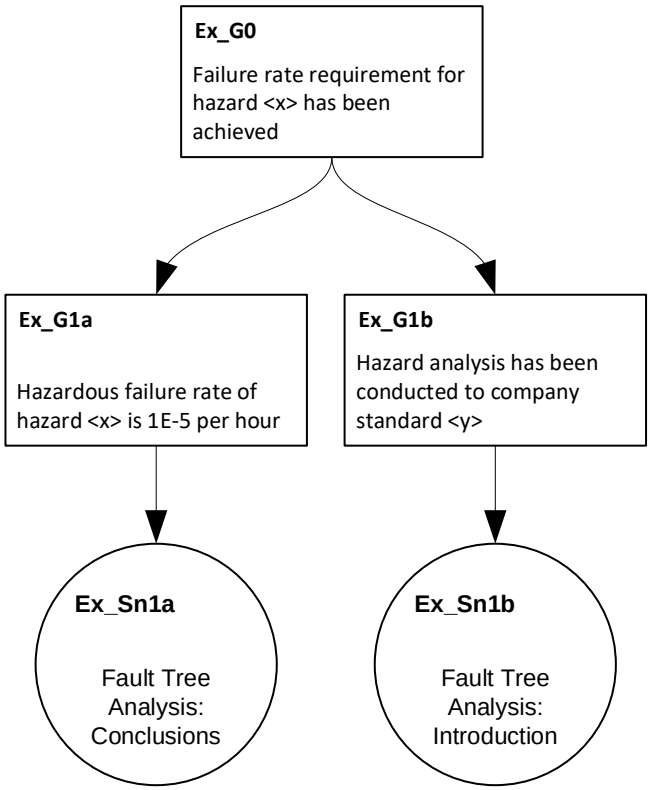


Figure 2:4-5 Adding a Higher-Level Goal

2:4.4.2 In considering how goal elements may combine to enable more abstract claims to be made, care needs to be taken to avoid jumping too quickly to the ultimate objective of the top goal, and it may be necessary to have a number of trial-and-error attempts at combining lower-level goals before a useful approach is found.

2:4.4.3 Goals should not be exclusively product-oriented – often, process evidence can be obtained from entities like FTA. This can demonstrate that the results of the approach used to create the FTA are trustworthy. Such evidence can hence be used to support a process-based strand of argument in the goal structure.

2:4.4.4 Note that the evidence assertion goal and supporting solution relating to the FMECA evidence has been omitted from the GSN fragment in Figure 2:4-4 – the same steps are required to complete that area of the argument. The author should not be pressured into manipulating evidence to support evidence assertions or goals that do not directly relate to it – the argument must be allowed to develop naturally.

2:4.5 Bottom-Up Step 4: Describing the Strategy for Goal-Decomposition

2:4.5.1 When deriving a parent goal that is supported by other goals, it can be helpful to describe how those goals support the claim made in the parent goal. Note that unlike the top-down process, the author will seldom have any choice as to how the goal to supporting goal decomposition is achieved. Figure 2:4-6 shows the addition of a strategy to describe the step between parent goal and its supporting goals:

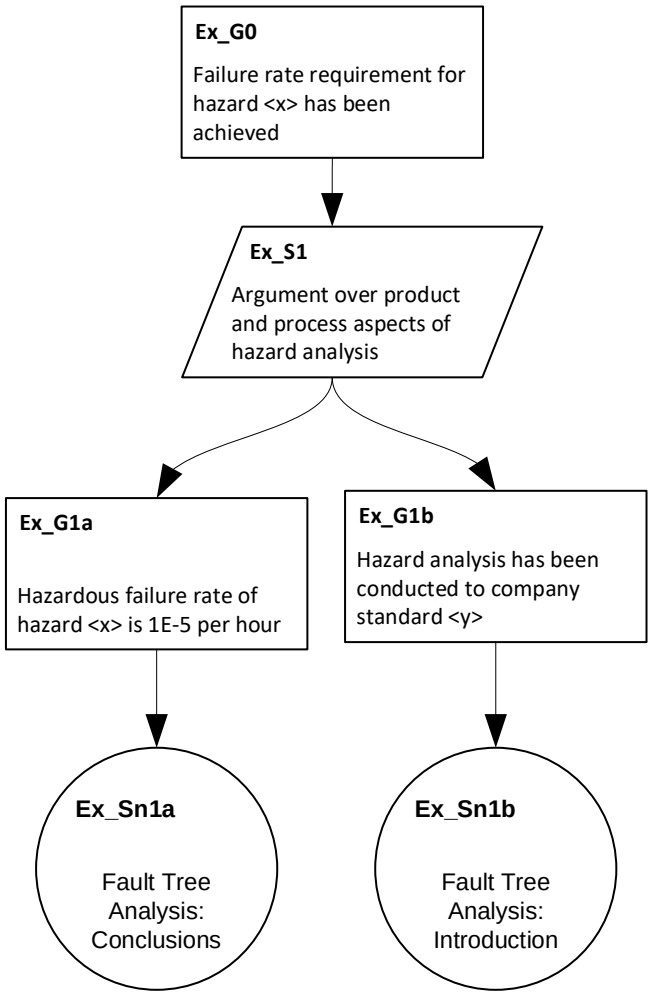


Figure 2:4-6 Describing the Strategy for Goal Decomposition

2:4.5.2 If the decomposition strategy is obvious, it may not be necessary to represent it explicitly as part of the goal structure. However, it is crucial that the author understand what strategy has been adopted in order to complete the following steps.

2:4.6 Bottom-Up Step 5: Adding Contextual Information

2:4.6.1 The creation of a goal structure from existing evidence may have elicited contextual information, including assumptions, definitions and references. Figure 2:4-7 shows the addition of contexts to the parent goal:

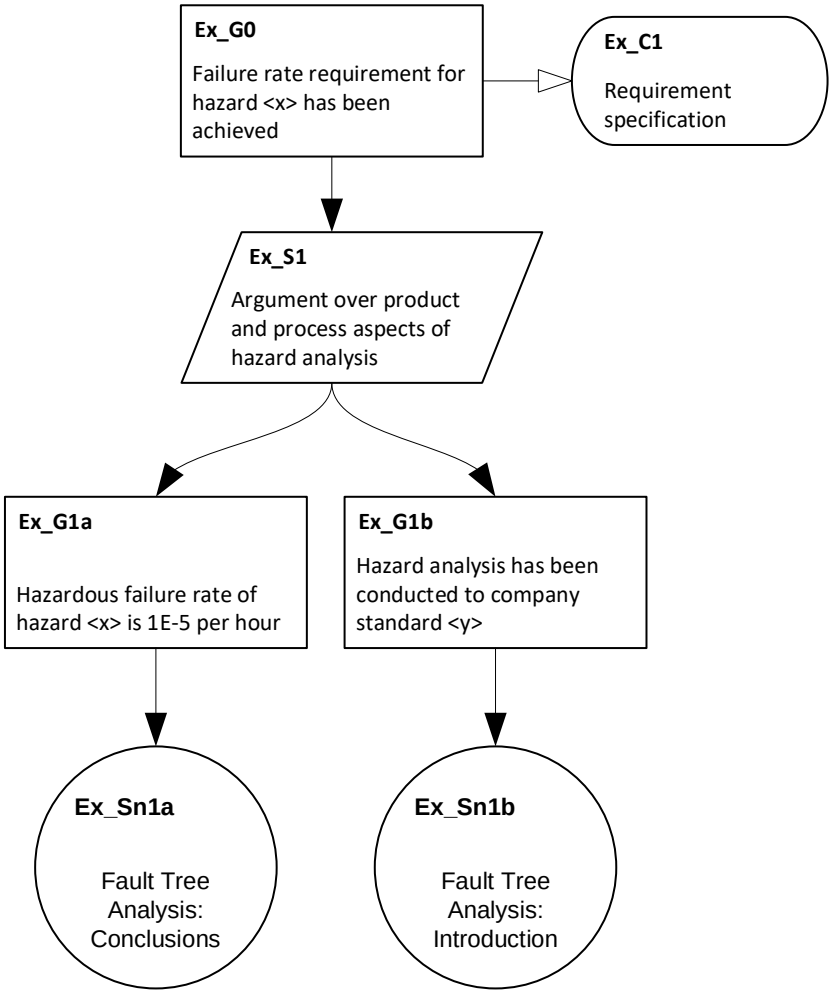


Figure 2:4-7 Adding Contextual Information

2:4.6.2 For example, an appeal to FTA evidence can provide a number of contextual items (not illustrated in the diagram above):

- An explicit system model which can be applied as a contextual reference in GSN, thus providing scope for the bottom level of evidence assertion claims made in the argument;
- Assumptions concerning system usage, e.g. number of hours per mission, number of operating hours per year;
- Assumptions about independence between elements of the system being modelled.

When developing the argument from the bottom up, these considerations can be useful to ensure completeness.

2:4.7 Bottom-Up Step 6: Checking Back Down the Structure

2:4.7.1 Each time a parent goal is created, the author should re-examine the supporting goals top-down, to check for adequate support of the claim made in the parent goal. This exercise should also extract the strategies used to make the inference between the supporting goals and the parent goal.

2:4.7.2 However the high-level goal structure is arrived at, it is recommended that the author make reflective top-down examination of the structure at each step. This should consider whether the supporting goals provide sufficient coverage of and support for the claim made in the newly created parent goal, and whether any assumptions or other context has been relied upon to make the inference step. The results of this evaluation may indicate that other supporting goals, solutions or context are required, or that the claim made in the parent goal needs to be rephrased.

2:4.7.3 For example, in the goal structure developed in Figure 2:4-3 to Figure 2:4-7, one result of this “check back down” step might be the identification of a requirement for operator competence to conduct FTA or a demonstration of absence of common causes.

2:4.8 Bottom-Up Step 7: Incorporating the Bottom-Up Goal Structure into a Higher (Top-Down) Argument

2:4.8.1 The bottom-up approach will rarely be used in isolation to form a complete goal structure. It is more likely that it will ‘join’ to a desired higher-level claim that is already understood to be a requirement of the associated assurance case.

2:4.8.2 Since the goal structure is developed from the existing evidence, the author should keep in mind where the argument is ‘aiming’ i.e. it should be written in such a way that it bridges the gap between a known argument claim higher up and the existing evidence. Figure 2:4-8 illustrates this connection:

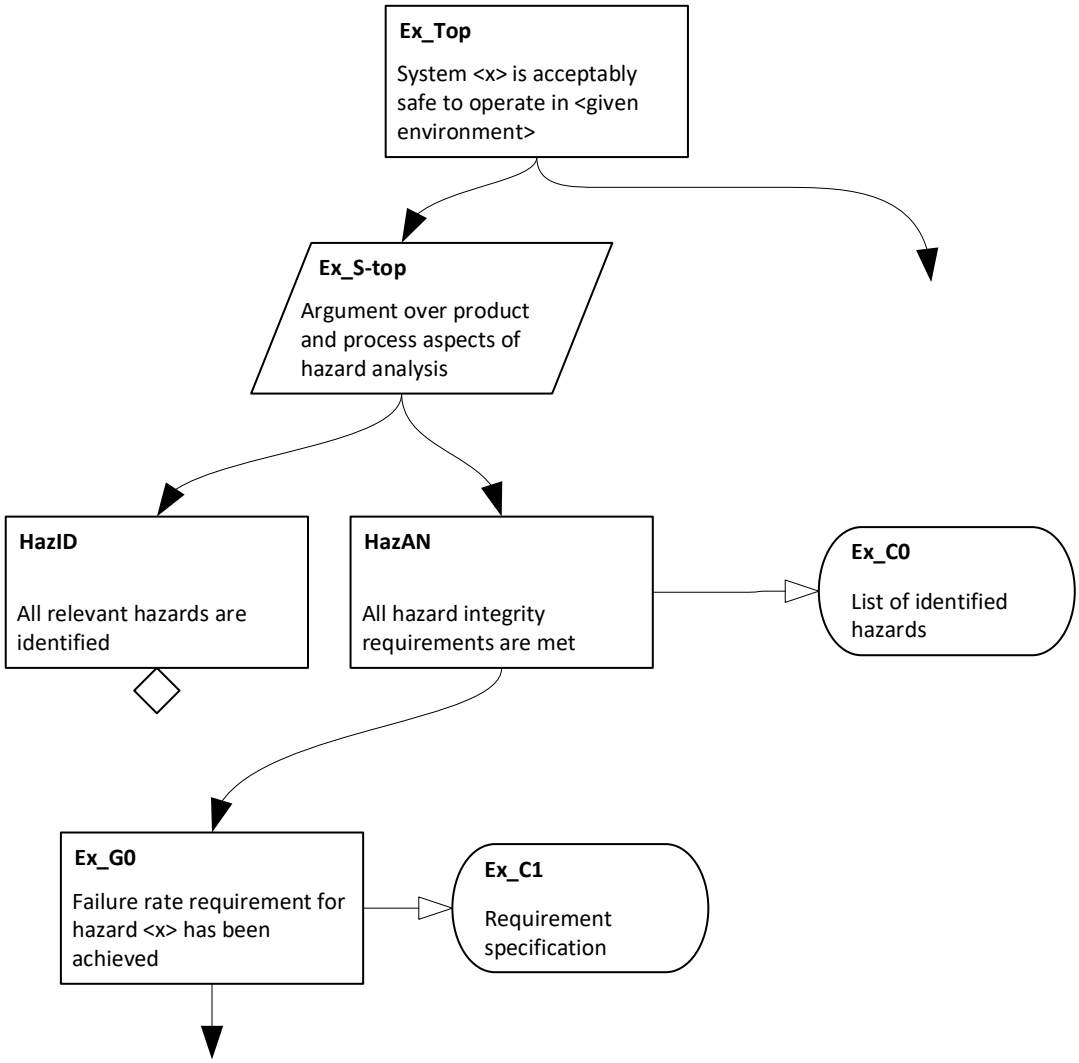


Figure 2:4-8 Joining the Bottom-Up Goal Structure to a Higher Fragment

2:4.9 "What if I Can't Convince Myself?"

2:4.9.1 When assessing the argument constructed from the evidence available, the author may realise that the evidence is inadequate to support the claims that have been made with sufficient confidence. The evidence might, for example, be incomplete, or might relate to a different version of the system from that addressed by the argument, or might be expressed in the context of assumptions which can no longer be held to be valid. In such cases, it is important that the author be honest about the limitations of the evidence, and scope the claims accordingly. Where possible, claims which are potentially undermined by shortcomings in one evidence item should appeal to more than one evidence item for support.

2:5 Avoiding Common Errors in Creating Goal Structures: Part 1- Language Issues

2:5.1 Introductory

2:5.1.1 The guidance presented in this and the following section (Section 2:6) is based on ‘real-world’ experience of the development of goal structures. It identifies some of the mistakes commonly made in argument development. Language-related problems are considered in this section, while Section 2:6 addresses difficulties in structuring goal-based arguments. Some of these pitfalls are specific to graphical approaches to arguments, while others arise from the use of argument techniques per se. Although the examples given below are taken from the safety domain, the problems identified and the guidance given apply generally to arguments of all kinds. It should be noted that, while the most commonly encountered issues have been identified, these two sections are by no means exhaustive.

2:5.2 Language Used in GSN Elements

2:5.2.1 In order to simplify the logic of the argument, it is important to state claims atomically, that is to ensure that each goal element contains only one claim. Where two parallel claims can be made – as, for example, in the statement “the design accommodates common-cause and common-mode faults” -, two goal elements should be used, to ensure that the logical structure of the argument can be expressed clearly.

2:5.2.2 The statements made in GSN goal elements capture the claims made in the argument. They should be expressed in the form <noun-phrase><verb-phrase>. The noun-phrase identifies the subject of the claim – i.e. the thing with which the statement is concerned. The verb-phrase defines a predicate – it serves to make some assertion about the subject.

2:5.2.3 Similarly, assumptions should be stated atomically in GSN.

2:5.2.4 GSN strategy elements record the approach used in structuring the argument. Strategies should not themselves form a necessary part of the argument: it should be possible to remove all of the strategy elements from an argument without affecting the logical flow of the claims being made. In order to focus attention on the function of strategy elements, it is useful for the author to introduce a summary of the argument

approach with a phrase such as “Argument by appeal to…”, “Argument by …”, “Argument across …”

2:5.2.5 The modular extensions to GSN introduce a few additional language considerations:

2:5.2.6 Module references must be unambiguously identified, and must therefore carry a module identifier. This identifier is used in away goal, away solution, away context and module elements. The module identifier must uniquely identify an argument module within scope of the overall argument framework. For clarity of the argument, the module element should carry a description of the nature of the argument contained within the argument module. The module description should be expressed as a noun-phrase.

2:5.2.7 The statement in away goal, away solution and away context elements should exactly match that in their referenced module counterparts.

2:5.3 The ‘Essay in the Box’

2:5.3.1 There is a tendency for the authors of GSN arguments to overload goals, strategies and solutions by writing lengthy summaries of the argument in a single element. This practice subverts the argument, since the resulting ‘essay in a box’ will typically contain several claims – about the system and/or the evidence items – which cannot be adequately supported, contextualised or elaborated in a goal structure.

2:5.3.2 In general, the textual element of GSN arguments should be kept as brief as possible, though the statements made in strategies, justifications, assumptions and textual definitions should be expressed using as much detail as is necessary for the reader to understand the nature and structure of the argument. The ‘essay in the box’ can be avoided by adhering to the following principles:

- **Atomicity** – The statements made in GSN goal, context and solution elements should be stated atomically. In other words, a single node should contain exactly one claim or reference. The use of more than one verb-phrase in a goal-statement often indicates that the goal contains multiple claims, as does the existence of more than one noun-phrase preceding a single verb-phrase. Where context or solution elements contain more than one noun-phrase, this may indicate that they contain more than one reference.

- **Allowing the goal structure to carry the argument**. When developing an argument, it is important to remember that each of the elements in the goal structure performs a specific role in structuring the argument: the 'argument' is the entire GSN structure, taken as a whole. It is therefore important that the text in GSN elements reflect the logical function for which the element was designed. Goals should only contain claims, solutions should only refer to evidence and strategies should only summarise the argument approach. Particular care needs to be taken to ensure that strategies do not restate – or, worse, redefine – the argument process when it is clear from the goal structure. In such cases, strategies can safely be omitted. Similarly, it is important not to make goals do the work of the argument: where the relationship between goals at different levels in the decomposition is not clear, a strategy should be inserted in the goal structure to explain this. Where the argument requires that a claim be made about the nature of the support a solution provides for a goal, this should not be stated as part of the solution. Rather, the claim should be stated as a goal to which the evidence item provides a direct solution.
- **Allowing contexts to act as references**. context and solution elements in GSN should provide references to artefacts stored elsewhere. A single noun-phrase (perhaps accompanied by a further reference to the location of the evidence) should be sufficient to identify these artefacts. It is not necessary to summarise the content of the artefact in the GSN node.

2:5.4 Ambiguity

2:5.4.1 'Ambiguity' is defined as "the capability [of a word or phrase] of being understood in two or more ways" (Shorter Oxford English Dictionary, 2007). Two types of ambiguity are commonly distinguished: lexical and syntactic.

2:5.4.2 In cases of 'lexical' or 'semantic' ambiguity, the ambiguity arises from multiple meanings inherent in a single word or phrase. It is worth noting that language dialect considerations may come into play here. The requirement "A warning light shall flash momentarily" would mean something rather different to a speaker of US English (who would interpret 'momentarily' to mean "in a moment, presently") than it would to a speaker of British English (who would expect the light to flash only once, for a short time).

2:5.4.3 In cases of ‘structural’ or ‘syntactic’ ambiguity, the grammatical structure itself allows for multiple correct interpretations. The claim “System functional software requirements development is acceptably safe”, for example, has at least five correct interpretations. The subject of this claim might be (i) the software functional requirements, (ii) the system functional requirements, (iii) the system requirements allocated to software, (iv) the interface between system and software or (v) the development of the requirements. One source of structural ambiguity concerns the scope of qualifiers – principally adjectives and relative particles – in clauses containing two or more nouns. It is often unclear which of the nouns the qualifier is attached to. ‘Limiter’ words (such as ‘only’, ‘also’ etc.) can lead to ambiguity when placed immediately before the main verb in a clause. Expressions of this kind can be easily avoided by placing the limiter word before the word which it seeks to constrain.

2:5.5 Vagueness

2:5.5.1 Certain words routinely used in arguments are essentially meaningless, unless they are clearly defined in the context of use. Where any of the following list of words is used in a claim made in a GSN element, a context should be added, specifying the precise meaning, in verifiable terms: ‘abnormal’, ‘appropriate’, ‘approximate’, ‘effective’, ‘early’, ‘easy’, ‘envelope’, ‘flexible’, ‘friendly’, ‘generally’, ‘late’, ‘normal’, ‘often’, ‘timely’.

2:5.5.2 Care should also be taken to avoid the danger of overstatement when using expressions including ‘all’, ‘any’, ‘each’, ‘every’, ‘typical’ and similar words. The author should consider whether so strong a claim is in fact justifiable.

2:5.5.3 In the same way, writers should avoid ‘blanket terminology’, where a single word is used to represent several instances or groups of things. Does the term ‘software’, for example, refer to a particular application, an entire embedded system, or computer programs in general? Particular care should be taken when writing GSN structures, since there is an assumption that the scope of terms is inherited from statements at a higher level. In practice, however, a given term may be subtly redefined at successive stages in the argument – the ‘software’ example above is a likely case in point. It may be necessary to introduce qualifiers for clarification purposes, e.g. to talk about ‘application software’ at one level and ‘control software’ at another. Context could also be used to define what is meant by ‘software’ but care

should be taken to avoid ambiguity and overload for the terminology. It is also important to ensure that the redefinition does not leave higher level claims inadvertently unsupported such as by supporting higher level ‘software’ related claims with ‘application software’ redefined context but failing to address other sub-contexts such as ‘operating system software’.

2:5.5.4 An overly qualified understatement can also lead to a claim which is unhelpful, in terms of developing the argument. For example, a claim that ‘some hazards have been identified’, while true – and easier to support than a more general claim – is largely uninteresting, in terms of developing a convincing assurance argument.

2:5.6 Oversimplification

2:5.6.1 Another potential danger in defining goals – particularly at the top level of the goal structure – is oversimplification of the claim made in the goal. Oversimplification can lead to vagueness, or to the argument’s appearing to make too great a claim for the system under discussion. For example, a top-level goal stated as “all hazards have been mitigated” could be regarded as an oversimplification, if it is true only that all of the major hazards have been mitigated.

2:6 Avoiding Common Errors in Creating Goal Structures: Part 2 – Structural Issues

2:6.1 Jumping Ahead

2:6.1.1 One of the potential dangers associated with defining the top goal of an argument is ‘jumping ahead’, i.e. stating a claim which supports the overall objective of the argument, rather than actually stating the objective itself. For example, the author of an assurance argument might put forward the top-level claim “Interlocks fitted to machinery”, rather than “risk associated with hazard X has been reduced”. The result is that higher-level justification of the mitigation strategy is omitted from the argument. If in doubt as to the level at which to address the top-level claim the author should consider what is the most fundamental objective relevant in the context. In this case, it is probably more important that the reader understands that the risk has been reduced than how it has been reduced.

2:6.2 Erroneous Use of Context

2:6.2.1 In GSN, context elements should not be used to refer to information which is intended to support the validity of a claim. Such information is evidence for the truth of the claim made in the GSN goal, and as such should be represented using a GSN solution element. Figure 2:6-1 illustrates this incorrect use of a GSN context element to support a claim made in a goal:

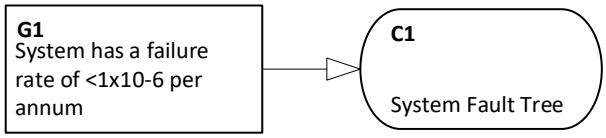


Figure 2:6-1 Incorrect Use of Context (as a Solution)

2:6.2.2 Here, Context C1 is incorrectly associated with Goal G1 as evidence offered in support of the failure rate claim made in the goal. The correct way to represent this relationship is to associate the System Fault Tree with Goal G1 as a GSN solution.

2:6.2.3 Context elements are sometimes used where a GSN assumption or justification may be more appropriate. In Figure 2:6-2, for example, the statement "System X has no common-mode failures" would be more appropriately rendered as an assumption rather than as context:

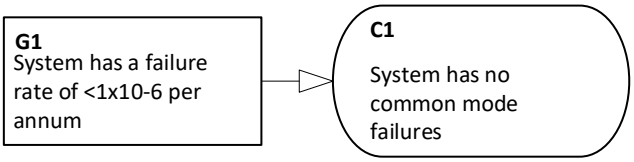


Figure 2:6-2 Incorrect Use of Context (as an Assumption)

2:6.3 Erroneous Use of Strategies

2:6.3.1 In GSN, strategy elements are intended as a description of the argument approach which has been carried out to relate claims stated at different levels of detail. They should therefore be expressed from the perspective of the argument, rather from that of the system, the design activity, testing or analysis. For example, the strategy "Interlocks used" should be phrased "Argument by appeal to the use of interlocks", to focus the reader's attention on the argument process, rather than on the design of the system.

2:6.3.2 Another common mistake is for GSN strategies to be deployed as 'load-bearing' elements, i.e. elements carrying some aspect of the argument, rather than

simply describing how it is structured. In such cases, strategies contain statements which are actually claims in the argument. Such claims are either made explicitly as part of the strategy, though they are often merely implied. Claims contained in strategies, rather than in goals, cannot be properly supported by the subsequent goal structure, and will therefore remain undeveloped, and unacknowledged, in the argument.

2:6.4 'Leaps of Faith'

2:6.4.1 Authors of arguments – whether they use words, mathematics or a graphical representation – often fail to persuade their audience simply because they fail to 'lead' the audience sufficiently. In other words, authors commonly assume that their audience is following the logical path they are setting out in establishing their conclusion, while in fact the audience has 'lost the thread'. The error here is in making too large an 'inductive leap' between claims, or between a claim and the evidence which is offered in its support. The error is akin to that in which a mathematician fails to 'show their working' between steps in a proof, thus making it difficult to see how they reached an interim stage or a solution.

2:6.4.2 In arguments represented using GSN, this error occurs when an author leaves too large a gap either between goals at different levels or between a goal-statement and a solution. In the first case, the inductive leap results in a lack of clarity as to how the lower-level goal relates to its parent. In Figure 2:6-3, for example, it is difficult for the reader to see the relationship between G1 and G8, since the reasoning by which inclusion of a safety cage justifies a claim of acceptable system safety is not clear:

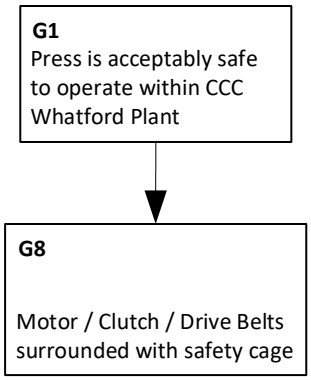


Figure 2:6-3 An Inductive Leap

2:6.4.3 In order to ensure that the reader can follow the logical thread of the argument, the author should add some additional goals between G1 and G8, to serve as ‘stepping stones’ the reader can follow as shown in Figure 2:6-4:

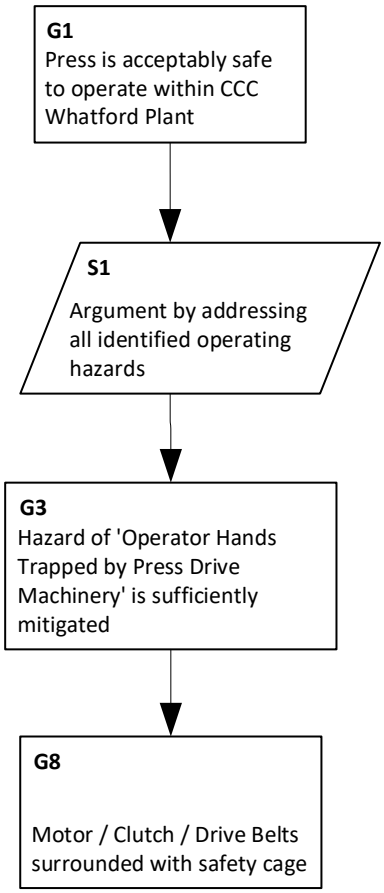


Figure 2:6-4 Intermediate Goal as a 'Stepping Stone'

2:6.4.4 Another common error is for the author to attempt to ‘close out’ a goal prematurely by direct reference to evidence in a way that will not be easily understood by the reader. For example, consider the solution element provided in Figure 2:6-5:

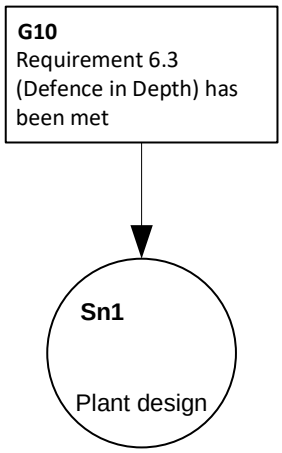


Figure 2:6-5 'Jumping' to a Solution

2:6.4.5 In this example, it is highly likely that, because the relationship between the requirement and the plant design has not been adequately explained, a potential reader may be confused or unconvinced as to how the claim made in Goal G10 can be inferred from the evidence referred to in Solution Sn2. In such cases, additional intermediate goal statements should be inserted between the goal and the solution (i.e. the goal should be decomposed further before reference to direct evidence). For example, Goal G10 could first be supported by other goals explaining how the defence-in-depth principle has been met in the design.

2:7 Evaluating Goal Structures: A Step-by-Step Approach

2:7.1 Introductory

2:7.1.1 Goal structures are used to provide assurance that the top claim(s) in an argument can reasonably be taken to be supported by the lower-level claims and evidence, with an appropriate degree of confidence. By their nature, arguments are often subjective and have many stakeholders e.g. developer, user, auditor, regulator, etc. This section provides a systematic approach to the review of arguments presented in goal structures, and also provides guidance on assessment of the level of assurance the argument provides.

2:7.1.2 The role of review within the argument development lifecycle is discussed in Section 2:7.2. Typical problems encountered during the review of assurance cases are outlined in Section 2:7.3. Against this backdrop, Section 2:7.4 presents a staged

argument review process which ranges from identifying simple problems of argument comprehension to the more difficult challenges of argument criticism and defeat.

2:7.2 The Role of Review in the Lifecycle

2:7.2.1 In terms of risk to the project, staged review throughout the project lifecycle is desirable. If there are problems with the arguments and evidence being presented, it is desirable that this be discovered as early as possible in the lifecycle. For example, early review of the strategy adopted in an assurance argument could be very useful.

2:7.2.2 The most compelling staged reviews will involve representatives from those responsible for acceptance and all other key stakeholders. It is often not possible to get confirmation that an interim conclusion is acceptable. Instead, the concern when involving these stakeholders is to obtain a 'non-negative' response – i.e. to know that, as it stands, the case does not contain any serious flaws in reasoning or weaknesses in evidence.

2:7.2.3 Even when it is impossible to involve acceptance authorities in interim review activities, self-review by the organisation preparing the argument is an extremely useful activity. Often the most difficult people to convince of the assurance of a system are those who know it best! Self-review requires the involvement either of people within the organisation who have maintained some independence from the development of the assurance case or of individuals capable of imaginative role-play along the lines of "If I were responsible for acceptance, what would I find unconvincing about this argument?"

2:7.3 Problems Commonly Experienced in Reviews

2:7.3.1 A key difficulty reported by those regularly involved in reviewing and accepting assurance cases lies in discerning the elements and structure of the argument being presented. The first step in reviewing any argument is first to be able to identify the argument being put forward. Too often, reviewers are required to perform 'industrial archaeology' to uncover the arguments and evidence. This difficulty can often lead to rounds of review comments primarily concerned with the presentation, rather than the structure or content, of the argument.

2:7.3.2 Once the argument has been uncovered, there can be further difficulties. For example, it can be very easy for the author to assume too much knowledge on the part of the reader. It will usually be the case that the people responsible for reviewing

the assurance case will have less knowledge of the system under scrutiny than does the author. It can be easy to make 'leaps' over stages of reasoning which appear obvious, or to refer to system concepts or to use terminology or acronyms which are confusing for the uninitiated reader.

2:7.4 A Staged Argument Review Process

2:7.4.1 Figure 2:7-1 illustrates a staged approach to the review of assurance case arguments, derived from [5]:

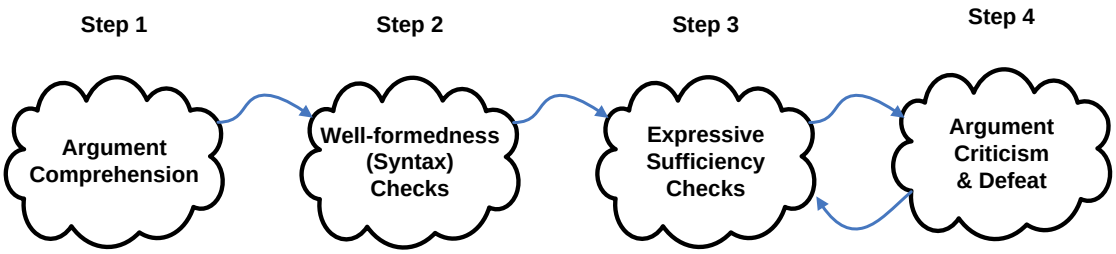


Figure 2:7-1 Staged Argument Review Process

2:7.4.2 Reviewing assurance case arguments can be thought of as comprising the following four steps, at least:

1. Argument comprehension;
2. Well-formedness checks;
3. Expressive sufficiency checks;
4. Argument criticism and defeat.

2:7.4.3 These steps are presented here both in order of necessity (e.g., the well-formedness of an argument cannot be checked before its structure is fully comprehended) and the order of difficulty. The latter stages require more intellectual effort and domain knowledge than do the former.

2:7.4.4 Given that the steps are presented in order of necessity, where a step cannot be completed satisfactorily, there may be little point in proceeding to the next step. For example, if it becomes clear in step 2 that the argument is not 'fully connected', there is little point in moving on to consider its expressive sufficiency (step 3). Argument review can require considerable expertise and effort. It would therefore be sensible to halt the process if insufficient information at any one step appears likely to create cascading problems for later steps. For example, an argument may simply

appear to be weak (picked up in review step 4) because it has not been made sufficiently clear (the concern of step 3).

2:7.4.5 Sections 2:7.4.1 to 2:7.4.4 describe the activities and concerns of each of the four steps of the review process.

2:7.4.1 Step 1: Argument Comprehension

2:7.4.1.1 In order to assess the argument, it is first essential that the reviewer understand the argument being presented. This step involves attempting to identify the key claims, strategies, assumptions, context and evidence presented in the assurance case. Where the argument has been documented in GSN, this step should require minimal effort and would comprise checks that the notation has been used in accordance with the normative description in Part 1: of this Standard. For example, checks can be made to ensure that phrases within strategy elements do indeed express argument approaches, rather than intermediate claims. This step will help to identify and weed out superficial arguments – i.e. structures which have been constructed using GSN but which do not contain valid claims or arguments.

2:7.4.2 Step 2: Well-Formedness Checks

2:7.4.2.1 It is possible at this stage to identify structural errors in the argument under review. For example, circular arguments (in which the premises of the argument depend in some way on the conclusions of the argument) are rarely considered acceptable. At this stage, it may be possible to identify claims for which no supporting argument or evidence has been presented. Conversely, there may also be items of evidence whose role in the argument is unclear.

2:7.4.2.2 Depending on how late in the argument's development the review is being conducted, it may be expected that the argument be 'fully connected' – i.e. that there are no disconnected fragments of argument whose relationship to the overall argument is unclear.

2:7.4.2.3 Since checks carried out at this stage are essentially straightforward and relate simply to the syntax and structure of the argument, it may be possible to provide tool support to perform some of them automatically.

2:7.4.3 Step 3: Expressive Sufficiency Checks

2:7.4.3.1 The purpose of this step is to assess whether the arguments have been expressed sufficiently for the argument to be fully understood. Often, elements of an argument can be implicit. The purpose of a strategy element in GSN is to explain the relationship between claims made in a parent goal and those in the goals supporting it. Explicit documentation of strategies is useful wherever this relationship is unclear. At this stage in the review process, it may be felt that further explanation of the inferences within the argument is required before any further review is carried out.

2:7.4.3.2 Equally, it is possible to add references to contextual information in GSN wherever the meaning of a goal-statement or strategy is unclear. In this review step, it may be necessary to demand that further context be defined before any further review can take place. This step is concerned with elements which may be missing from the context of the argument and whose absence prevents our gaining a full understanding of the argument.

2:7.4.4 Step 4: Argument Criticism and Defeat

2:7.4.4.1 Assurance arguments are generally inductive. The absolute truth of the conclusion cannot be established with certainty. Rather, the probable truth of the premises is passed through to the conclusion. In evaluating an inductive argument, it is necessary to establish its overall sufficiency: are the premises of the argument, taken together, strong enough to support the conclusion(s) being drawn?

2:7.4.4.2 The sufficiency of the relationship between premises and conclusion of the argument can depend on a number of attributes:

- **Coverage** – to what extent does the argument and/or evidence presented cover the conclusion? For example, a conclusion regarding all hazards which presents evidence only for a subset of the known hazards has a potential problem of coverage.
- **Dependency** – the level of assurance offered up by multiple forms of evidence or strands of argument may not be so convincing if they are not truly independent. For example, on inspection, two forms of evidence may both be found to use a common, flawed model of the system as a starting-point.
- **Definition** – it could be considered undesirable to over-constrain or under-constrain the argument or the evidence being presented. For example, an

argument of safety that is assured only for a narrowly defined operational context (e.g. “The system is safe on Tuesdays”) may be considered insufficient for the purpose of approving safe operation of the system.

- **Directness** – to what extent does the argument or evidence directly address the conclusion being sought? Against a specific product claim, process evidence can be regarded as ‘indirect’. Indirect arguments are often considered unconvincing in isolation.
- **Relevance** – how relevant is a particular piece of evidence or line of argument to the conclusion being sought? An argument that “the System is safe” because “the sky is blue” suffers from a problem of relevance. Although this is an extreme example, more subtle problems of relevance can exist. For example, the claim that a later version of a software item satisfies a requirement based upon test evidence concerning a previous version can present a problem of relevance.
- **Robustness** – how susceptible is the argument to changes in the evidence and claims arising from this? For example, consider an argument where an objective is considered to be ‘just’ satisfied, as opposed to one where the objective is exceeded by some margin. The latter would be considered by many to offer a greater degree of assurance, all else being equal. Alternately, where an intrinsically pessimistic assessment shows that a requirement has been satisfied (albeit only just), this may be considered more persuasive than an assessment based on a more optimistic approach which shows a greater margin of satisfaction.

2:7.4.4.3 When providing feedback from this step in the review process, it is advisable for the reviewer to be as specific as possible in identifying the problems present in the argument. Shortcomings noted against any of the above criteria are likely to indicate that an argument is insufficient. The author is likely to find a comment that there is a problem with “lack of coverage” more useful than a ‘blanket’ criticism like “insufficient argument”.

2:7.4.4.4 It is important to recognise that criticisms of the argument at this stage could simply relate to weaknesses of expression (the concern of step 3).

2:7.4.5 Step 5: Auditing the Evidence

2:7.4.5.1 There is a requirement incumbent on the assurance case review process to audit the evidence presented in support of the argument. The reviewer should ensure that the items of evidence referred to be the argument actually exist and that they actually support the claims of the case as presented. For example, if a claim is made that “All hazards have been closed out in the hazard log”, review of the hazard log should demonstrate that this is true.

2:7.4.5.2 In the abstract, the evidence (as referenced) may support the arguments as stated. However, if an evidence item is not considered sufficiently trustworthy, the argument may be undermined. In law, the concept of ‘integrity’ of evidence is used (especially in the case of forensic evidence). For example, if the evidence collection and analysis process cannot be assured, evidence can be ruled inadmissible or of reduced evidential weight.

2:7.4.5.3 For assurance cases, there are a number of possible factors to consider when assessing the integrity of evidence:

- **‘Buggy-ness’** – how many ‘faults’ are there in the evidence presented? The more mistakes revealed in evidence during a review, the less confidence the reviewer is likely to have in the evidence.
- **Level of Review** – has the evidence been produced and thoroughly reviewed by suitably competent and experienced personnel? This principle is already enshrined in several safety standards; for example, RTCA/DO 178C [6] requires independent review of software items developed to high Design Assurance Levels (DALs).
- In the case of hand-generated evidence, the experience and competency of personnel can be regarded as essential backing evidence.
- In the case of tool-derived evidence, tool qualification and assurance are important issues. DO-178C [6] makes an important distinction between tools where the output forms part of the final delivered product and tools with an ancillary role in the development process.

2:7.4.5.4 A good assurance case cannot be selective in the arguments and evidence it presents. Facts not included within the presentation of the assurance case may

challenge the argument. It is necessary to be prepared to consider whether such facts exist.

2:7.4.5.5 Consideration of counter-evidence is one of the most difficult aspects of assurance argument development, due to the open-ended nature of the challenge. Extensive domain knowledge is required for a reviewer to know that there is something not presented in an argument, or that an alternative interpretation of the evidence is valid (and further domain knowledge is required to establish which of several possible interpretations is most persuasive in the context). The reviewer's knowledge can challenge the argument in two ways: rebuttal and undercutting[4].

2:7.4.5.6 Rebuttal describes the situation where evidence exists that allows you to reach a conclusion counter to one presented in the assurance case. For example, if the assurance case claims that "Failure Mode X has never occurred", rebuttal would be to provide support for the claim "Failure Mode X has occurred" by reference to supporting arguments and evidence (e.g. a previous incident report). Rebuttal describes a 'head-to-head' dispute between the claims of the assurance case and counter-claims that can be substantiated.

2:7.4.5.7 Undercutting describes a situation in which additional arguments and evidence are introduced which challenge the reasoning (especially the inferences) presented within the argument. For example, consider the following argument:

Premise: The vehicle is travelling at 80 mph

Conclusion: The driver is breaking the speed limit

An additional fact, that "the vehicle is travelling along a private road", challenges the inference. During the review process, it is necessary to consider whether there are circumstances in which the premises of the argument are true, but the conclusions are false. Given the nature of an inductive argument, it is theoretically always possible to introduce an undercutting argument which defeats an inference step. There is therefore a need to use undercutting with some judgement to avoid chasing an unattainable deductive argument.

[4] It is acknowledged that the use of these terms here and in Section 2:11 are different and this will be resolved at the next Version.

2:8 Argument Pattern Extension Guidance

2:8.1 Use of Patterns

2:8.1.1 GSN is generally used to articulate a specific argument, relating to a particular system. It can be helpful, however, to generalise the specific details of a specific argument into patterns of reusable reasoning, akin to software development patterns or tactics. GSN has therefore been extended to support abstraction.

2:8.1.2 Two forms of abstraction are supported:

- **Structural Abstraction**, which allows the generalisation of a relationship which exists between two specific instances of a GSN element-type into a relationship between classes (e.g. representing one-to-one and one-to-many relationships);
- **Element Abstraction**, which allows a distinction to be made between classes and instances of GSN element-types.

2:8.1.3 Structural abstraction allows generalisation of the structure of an argument. For example, it is possible to indicate that, in general, at least two out of five possible forms of argument must be put forward in support of a particular claim.

2:8.1.4 Element abstraction allows generalisation (or postponement of detail) of an element in the argument structure. For example, for a goal claiming a particular failure rate, it would be possible to say that, in general, the solution will be “Quantitative Evidence” without specifying whether this is specifically “Fault Tree Analysis” or “Markov Modelling”.

2:8.1.5 Section 1:3 above defines the GSN symbology introduced to support structural abstraction: the optionality and multiplicity annotations attached to the SupportedBy relationship, and the choice symbol, which is used to represent alternative lines of argument used to support a particular goal.

2:8.1.6 Multiplicity relations can be combined with choice relations. Placing multiplicity annotations on the ‘supported by’ symbols prior to the GSN choice symbol describes a multiplicity over all of the alternative choice relations. Placing a multiplicity symbol on individual choice relations (i.e. just prior to the sink) describes a multiplicity over that relation only.

2:8.1.7 It is useful to provide an annotation next to the choice symbol denoting the nature of the choice to be made – e.g. ‘1 out of n’ or ‘2 out of 3’.

2:8.2 Template Arguments

2:8.2.1 A particular approach to instantiation of pattern notation, called ‘Templates’, can assist in optimising the level of information provided visually when a highly repetitive argument structure is required. For example, consider the template derived from a previously published pattern (High Level Software Safety Argument Pattern) as shown in Figure 2:8-1. Software elements might contribute to several different hazards; this could be represented by a fragment which is replicated for every occurrence where software contributes, or through using this template as a single instantiable fragment alongside a table that provides the instantiating data.

2:8.2.2 Whilst both representations are technically correct and equivalent, the former may result in a distracting amount of information. Figure 2:8-1 shows the example template and Table 2:8-1 is an example of instantiation data for that template, presented as a table. In this example the table is incomplete as may occur during the development of an argument, but this would be invalid in a complete, published, argument.

2:8.2.3 It can be seen that care is required in the phrasing of the instantiable parameters to ensure that the resultant argument is clear whilst keeping the instantiation data structure simple. An illustration of the instantiated equivalent argument using part of the template table is shown in Figure 2:8-2.

2:8.2.4 The pattern from which the template in Figure 2:8-1 is derived is not a complete argument and leaves elements undeveloped, i.e. ‘Goal: sw contribution’. When using a template, all elements must be developed either within the template, elsewhere within the argument or in another module (e.g. using a ‘solved by contract’ decorator). It is also possible to use of further templates which address each instantiated element.

2:8.2.5 It is noted that, where the template argument contains non-instantiable elements, this may lead to the same element included multiple times if the argument were to be fully instantiated from the table. This may cause unnecessary duplication but does not break GSN rules provided they are indeed identical, and is likely to apply more to contextual elements such as assumption, justification and context.

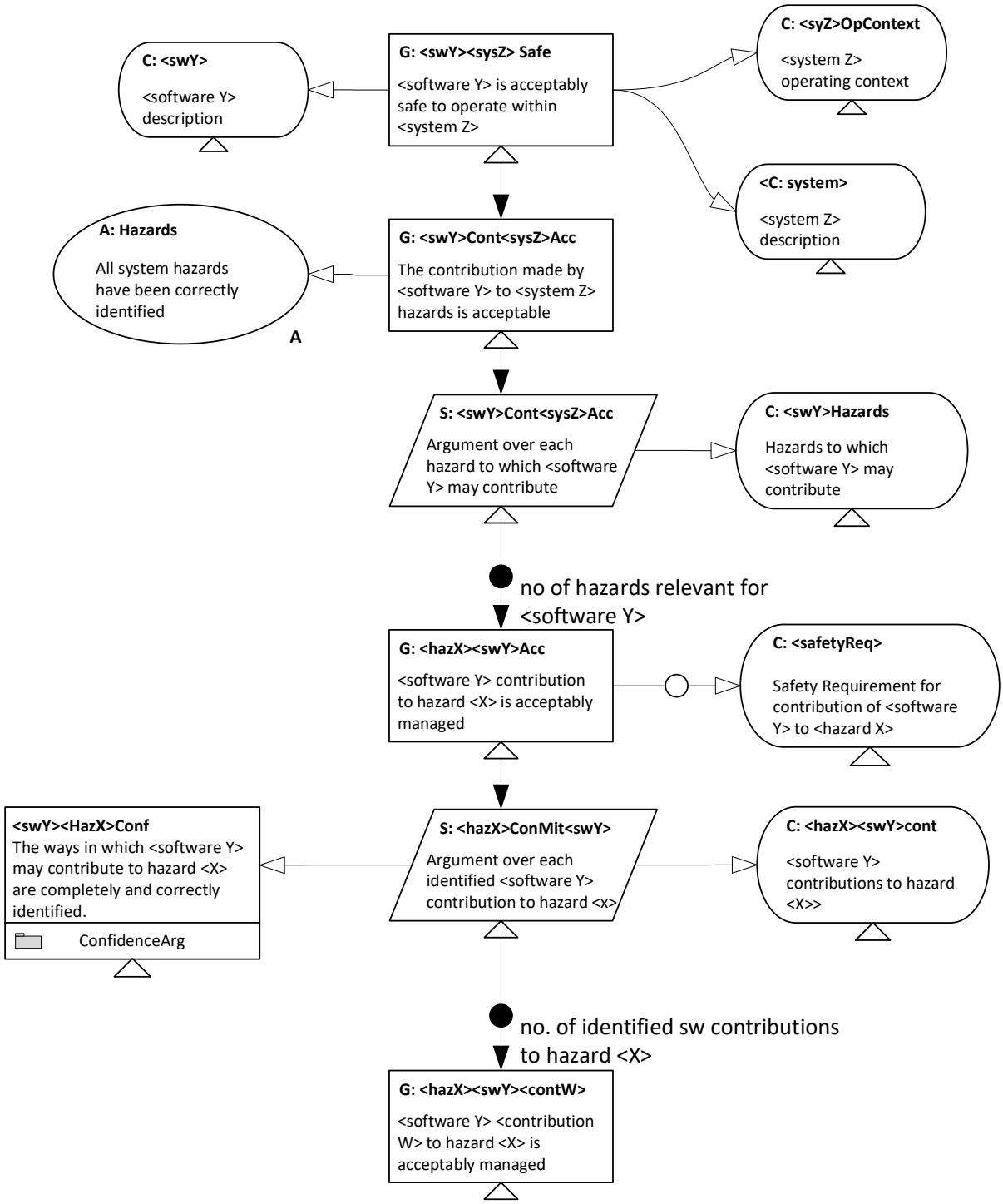


Figure 2:8-1 Example of a template argument

2:8.2.6 Optionality and multiplicity instantiation may be directed by description in the pattern description when the template is derived from a published pattern. Where optionality is included the instantiation data needs to explicitly declare where the option is not instantiated. In the example shown in Table 2:8-1 below, the optionality for c:<safetyReq> is not declared meaning the instantiation is ambiguous.

Table 2:8-1 Example Instantiation Data (partial table)

<system Z> [<sysZ>]	**<software Y> [swY]**	**<X> (hazard) [hazX]**	**<contribution W> [<contW>]**
Braking System [Brk]	Brake control Software [BrkCtrl]	Runway Excursion [RwEx]	TBD
		Runway Overrun [RwOR]	TBD
		Fire due to overheated brakes [OhtBrk]	TBD
Engine System [Engn]	Fuel Control Software [FuelCtrl]	Fuel starvation [Fstrv]	Fuel Distribution management [Fdist]
			Fuel Pump management [Fpump]
			Fuel Cut-Off Valve management [FCOV]
		Fuel Exhaustion [Fexh]	Fuel Monitoring management [Fmon]
			Fuel Jettison management [Fjet]

2:8.2.7 When using a table for instantiation data, the number of elements may be dissimilar per instantiation, for example, there are three hazards associated with the brake control software but only two with fuel control. The template author should consider whether this variation must be justified within the argument and ensure there is an instantiable element in the template for this justification, where required. This may be as part of a related process argument, for example. This is particularly important when the decomposition requires an argument to be made about sufficiency of support.

2:8.2.8 Where multiple instantiable elements are used in the same argument structure, care should be taken over whether additional columns are added to an overall table, or a single table is used per instantiation element, or some combination thereof. Options that result in greatest clarity to the reader are preferred. Particularly, it may not be clear whether a blank line in a table, due to the layers of instantiation, is intended, i.e. is not required in this instantiation, or is an unintended omission, which the author should have completed.

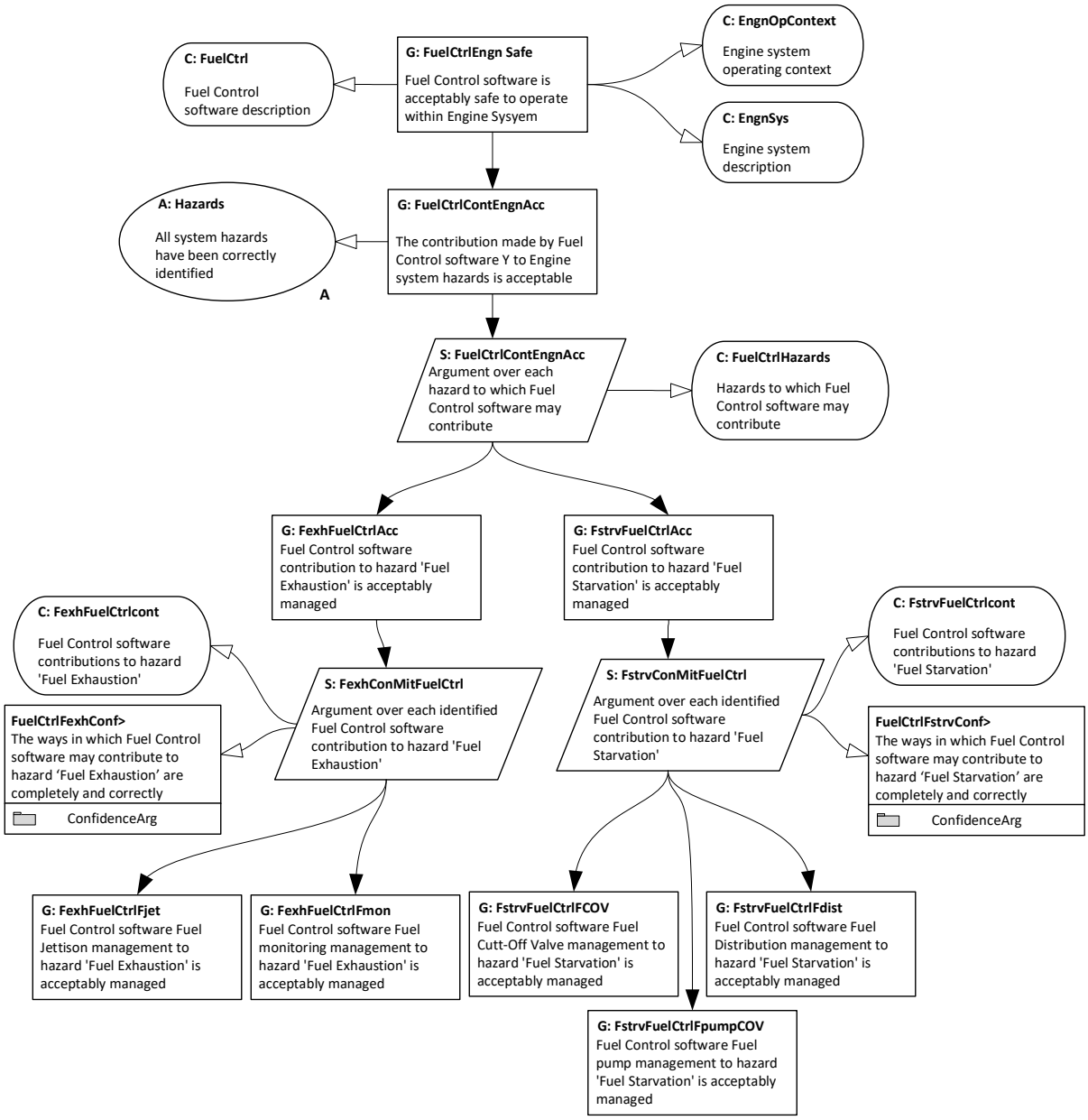


Figure 2:8-2 Example of the equivalent instantiated argument (partial)

2:9 Modular Extension Guidance

2:9.1 Introductory

2:9.1.1 Developing modular assurance cases is a powerful approach to addressing complexity in technical and commercial domains. Developing the architecture of the argument may be influenced by properties such as the resilience to change, reusability, commercial boundaries, etc. General guidance on a modular approach to assurance cases is provided in [8]. The following guidance in this section does not

repeat this general guidance, rather it provides guidance on GSN specific aspects of modular assurance cases.

2:9.2 Argument Visibility Considerations

2:9.2.1 The modular extension to GSN provides two key methods of providing visibility of the argument:

1. Share the argument structure, including all GSN elements and their relationships.
2. Share a published interface of the module containing only the subset of elements that are relevant.

2:9.2.2 The first of these is equivalent to providing the argument created using core GSN. In the case of a module containing both argument and other modules, this would involve sharing both the argument view and the architecture view. This provides full visibility of the argument and enables any reader to make reference to any element of the argument, though the author can indicate which elements are intended to be referenced by application of the public decorator.

2:9.2.3 The second approach can be used to limit visibility of the full argument in order to enforce privacy (e.g. for IPR reasons), to hide complexity (e.g. for comprehensibility) or to aid configuration management (e.g. by enforcing referencing to only the goals that are declared as public).

2:9.2.4 A combination of these two methods may be used, for example by sharing the full structure, or a sub-set of it, ‘for information only’ and sharing a selected interface or interfaces to allow the module to be related in a larger argument.

2:9.2.5 One of the decisions that any argument author will need to make is which elements should be made public. The author of the overall argument may prefer maximum visibility of the module contents and therefore promote making as many elements as possible public. The module author may not be responsible for the full goal structure and may wish to use modules as a mechanism to protect Intellectual Property by hiding the argument structure and only exposing the necessary elements through the module interfaces.

2:9.3 Use of ‘Away’ Contextual References

2:9.3.1 The modular extension provides for making away references to Justification elements. Whilst these are provided for completeness, it is anticipated that correct use of justifications means that they are specific to the connected element, or relationship between elements, that in the module in which they are declared, and it would therefore be unusual for them to be referenced from another module out of context.

2:9.4 Composing Arguments by Relating Modules

2:9.4.1 Modules can be composed into a larger argument by relating their interfaces through inter-module contracts. There are several options for representing contracts between module interfaces.

- In its simplest form, creating relationships between modules using ‘away’ references (such as away_goal, away_solution or away_context) creates an implicit contract. Ideally this should be supported by a tabular contract to ensure that all aspects of the modules’ interfaces have been addressed.
- The second is a tabular form, such as show in Table 2:9-1.
- Where an additional argument is needed to justify adequate support, (for example, where goals requiring support do not exactly correspond to a goal providing support), a GSN representation of a contract is recommended. An example form of a GSN contract module is show in Figure 2:9-1.

2:9.4.2 Regardless of which approach is taken to creating the inter-module contract, the following simple steps should be performed:

Step 1: Goal Matching: In this step an assessment is made of whether the public goals in one module provide full or partial support for the goal(s) requiring support in the other module. It is possible that A module provides support for a goal in another module, whilst also having a goal that is supported by the other module. This is acceptable providing it does not form a circularity in the argument e.g. if it involves separate branches in the argument.

Step 2: Consistency Checks: It is necessary to confirm the compatibility of the goals in the context in which they are declared. This includes all context that is in scope of the matched goals, which may include context ‘inherited’ from the argument not directly linked to the matched goal. The relevant context should be declared in the relevant module interface(s) by the module author.

Step 3: Cross-Reference Checks: Where cross-references are made between modules using 'away' references, a confirmation is required that these referenced elements exist in the referenced module. It is expected that this check will be performed against the published module interface, with the responsibility for the interface accurately representing the contents of the module resting with the module author. It is anticipated that suitable tool support will enforce the validity of these cross references.

Step 4: Support Sufficiency Checks: It is necessary to confirm that the goal matching provides a sufficiency of support for each supported goal. As well as the consistency and existential checks performed in steps 2 and 3, this involves confirming that the supporting goals are relevant and sufficient to address the supported goal. Where the matching of step 1 is not exact, this may involve finding other supporting goals that address the shortfall, or confirming that the supporting goal(s) are at least as 'strong' as the supported goal.

Table 2:9-1 Example Contract Table (partial)

Assurance Case – Inter-Module Contract			
Participant Modules			
Module A {interface x}, Module B {interface y}, Module C {interface z}, …			
Goals Matched Between Participant Modules			
Goal	**Required by**	**Addressed by**	**Goal**
Goal G1	*Module A*	*Module B*	*Goal G2*
…	…	…	…
…	…	…	…
Collective Context and Solutions of Participant Modules held to be consistent			
Context		**Evidence**	
Context C9; Assumption A2		*Solution Sn3, SN8*	
Resolved Away element references between participant modules			
Cross Referenced element	**Declared in (owner)**	**Referenced from**	
Goal 3	*Module B*	*Module C*	

2:9.4.3 The decomposition of the away goal, which is not permitted in other types of GSN module, is explicitly permitted in a contract module. In a contract module, the top away goal is used to reference a goal that is 'supported by contract' in its local module, i.e. it is explicitly not supported elsewhere than in the contract. The lower

away goal is providing support and references to a public goal in a separate module. The public goal referenced from the contract must be adequately supported in its own module for the contract to be valid.

2:9.4.4 The capture and recording of appropriate contextual information is perhaps the most critical aspect of using modular GSN effectively. Where only basic structuring is necessary, standard GSN guidance applies to recording context, assumptions, justification, etc. With indirect linking between modules and the potential to support change, a heavier dependence is placed upon capturing and justifying the compatibility of context, particularly where this is used for complex systems whose elements may be changed or reused in new systems. This is reflected in the use of away contextual element references and an explicit justification or argument about compatibility of the goals in the context in which they are declared as shown in the contract module example shown in Figure 2:9-1. By way of an example, if the top goal of the contract requires a property to hold true for 7 days per week, but the contextual information around the goal offering support constrains the support to Monday to Friday operations only, the contract is not valid, even if the argument and evidence supporting Monday to Friday operations is sufficient for the constrained case.

2:9.4.5 An alternative method for referencing the associated contextual elements is shown in Figure 2:9-2. This alternative method utilises the associations formed in each module interface to concisely reference the collection of relevant contextual elements without the need to explicitly show each element individually. It may be more convenient to use the alternative method when a significant number of contextual elements are relevant to the away goal.

2:9.4.6 Understanding what contextual information is sufficient, relevant and important to record is extremely difficult for complex systems. Checklists can be formed based on domain expertise and well-known examples of errors, such as mismatches between measurement units expected and received, but more subtle examples clearly exist, such as a mismatch in processor speeds between the target processor and the processor in the test environment used to generate supporting evidence for the argument. Careful thought and attention are needed and encouraged as part of the modular approach.

2:9.4.7 In the example shown in Figure 2:9-1, either or both DecompJust and ContextJust may require complex arguments so could be rendered as away goals and argued in separate modules contained within the Contract Module. Two different styles of referencing relevant context of the away goals in the contract are illustrated. Figure 2:9-1 illustrates the case where every contextual item is references as an away element, whilst Figure 2:9-2 shows a summary reference via the modules' interfaces.

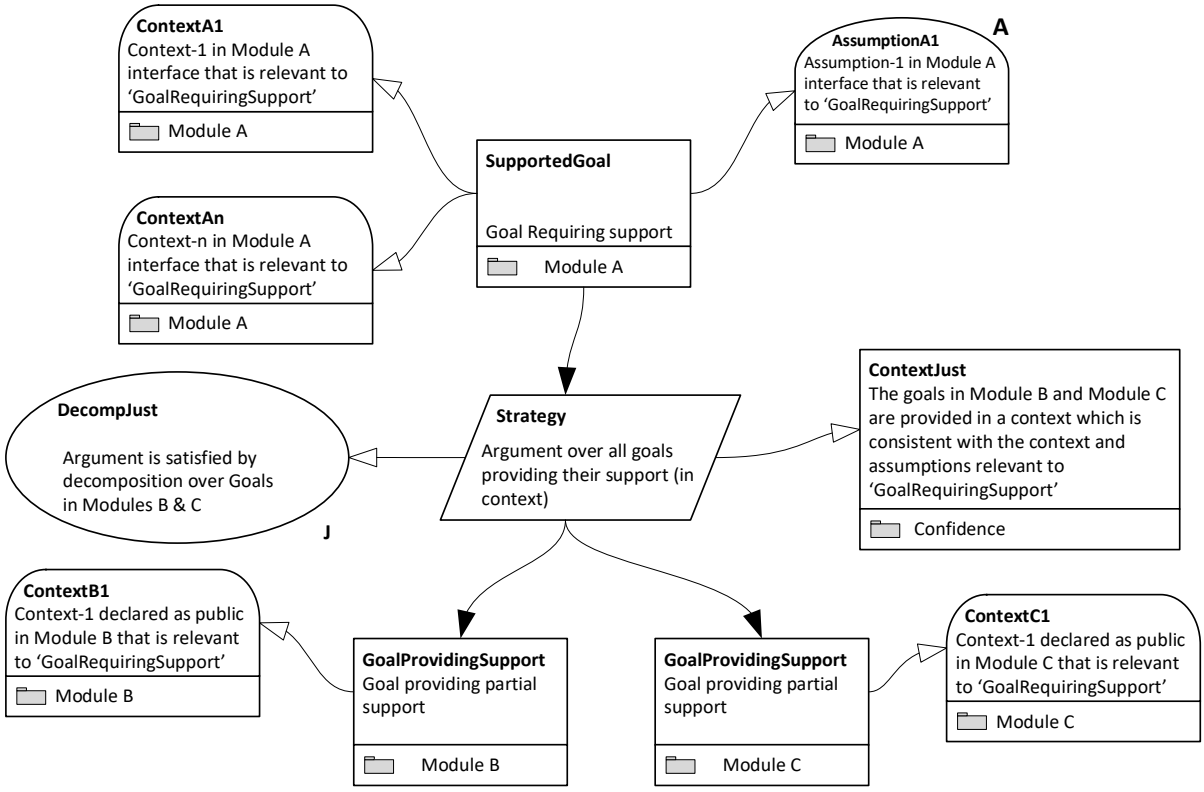


Figure 2:9-1 Generic Contract Module Example

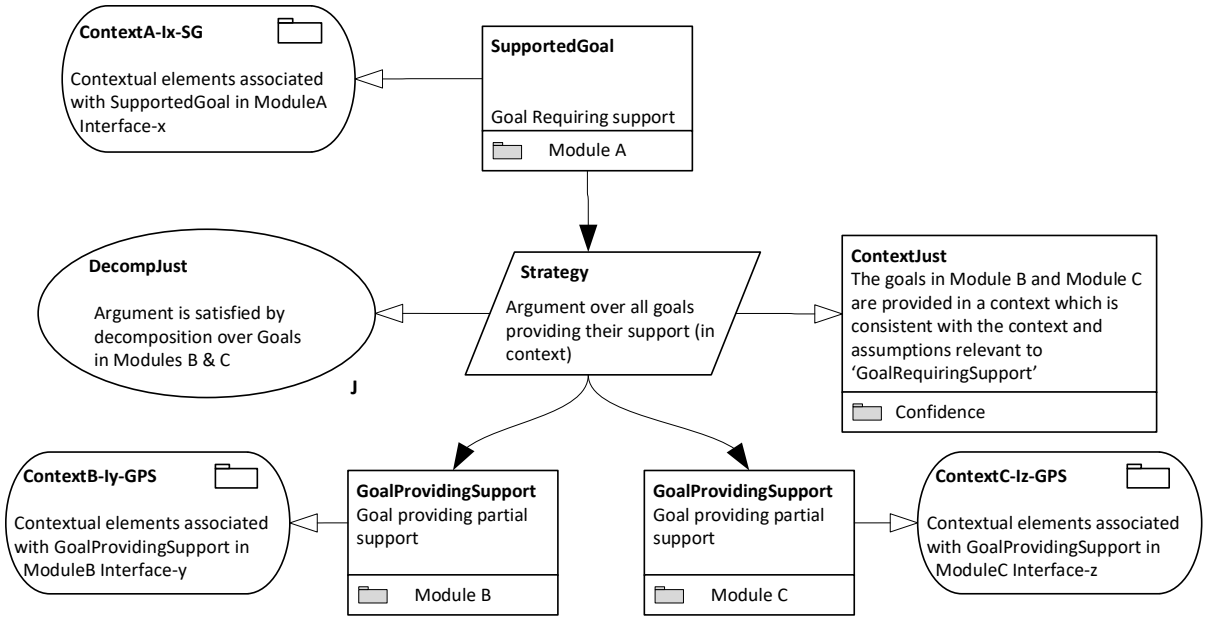


Figure 2:9-2 Alternative Contract Module Example

2:9.5 Use of Extended Views

2:9.5.1 This standard does not define normative requirements for extended views[5], however, it is recognised that it may be useful to illustrate goal structures in a single diagram that depict aspects of relationships between modules as well as the argument and other modules contained within those modules. Examples of such extended views are shown in Figure 2:9-3 and Figure 2:9-4. In these examples a different colour and non-standard shapes have been utilised to highlight the extended view aspects.

2:9.5.2 Figure 2:9-3 depicts selected argument content of related modules and a contained module within a contract module. It illustrates how the contract module argument (introduced in Figure 2:9-1) relates to the modules it supports. It shows that the confidence argument that supports the context justification for the support to module A, in this case, is addressed in an argument contained in a module within the contract module. It also illustrates how the top goal of the contract module relates to the goal requiring support in module A, and the bottom goals of the contract relate to the supporting goals in modules B and C.

[5] The modular extension defines two views:
- The Argument View, which shows the GSN argument elements and relationships as defined in Section 1:4.2
- The Architecture view, which shows the modules and contracts that the module contains and the inter-module relationships as defined in Section 1:4.4

2:9.5.3 Figure 2:9-4 depicts modules that contain other modules. This illustrates how an overall argument may be addressed by separating concerns into system, hardware and software aspects; and then how these aspects may be further addressed in their own modules. Note that in this example the higher-level modules show the specific interfaces being used in the relationships, whilst this information is omitted in the illustration of the contained modules to avoid clutter in the diagram.

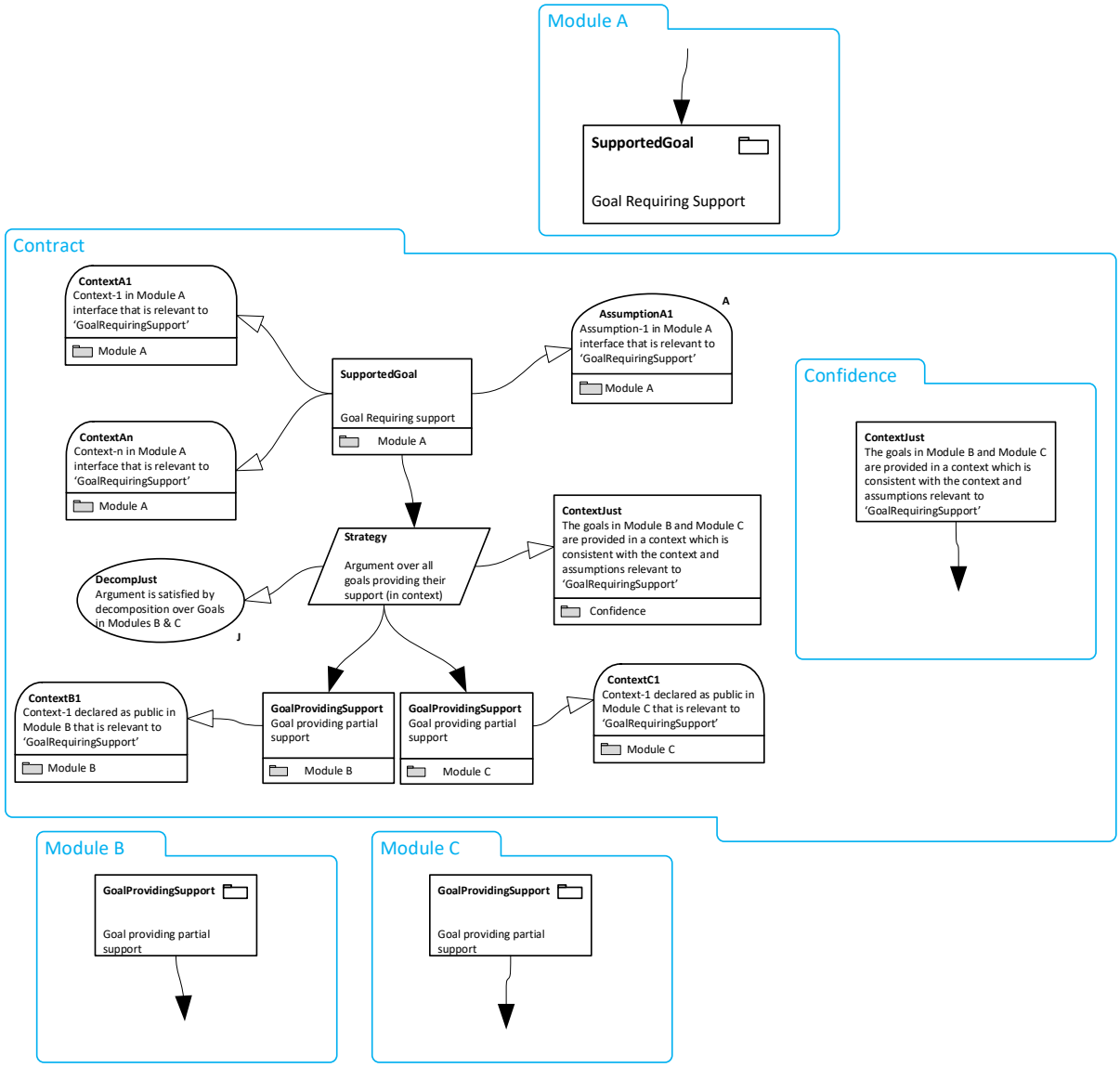


Figure 2:9-3 Extended View Example - Contract Argument and Related Modules

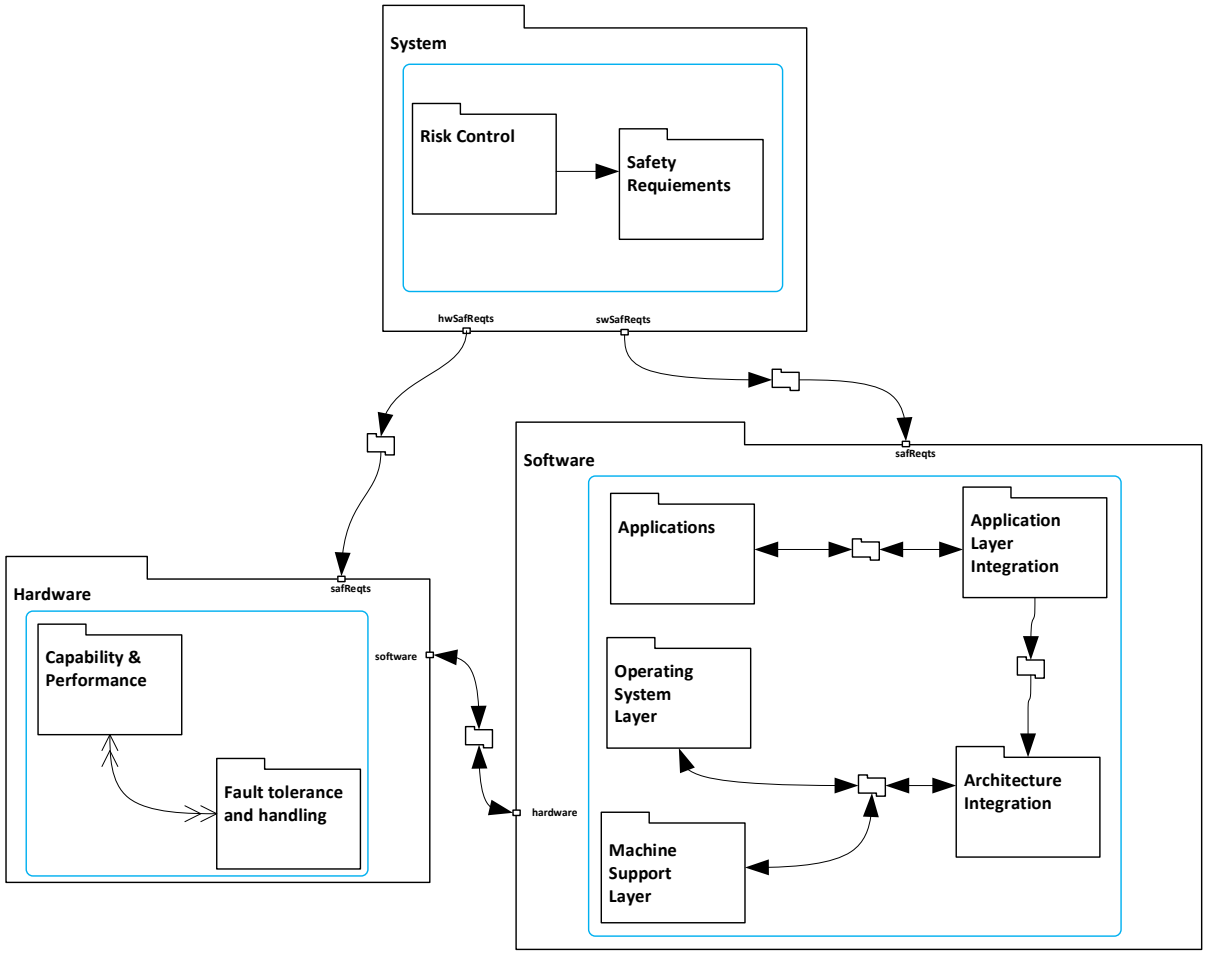


Figure 2:9-4 Extended View Example – Modules Contained Within Modules

2:9.6 Working with Module Interfaces

2:9.6.1 The concept of the module interface is introduced in Section 1:4.6. There are two key aspects:

a) Elements that are available to support other arguments located in other modules.

b) Elements which require support from arguments located in other modules.

2:9.6.2 All elements that appear in an interface of the module are public and should be rendered with the public decorator in all views. All public elements are available to reference from any argument that has visibility of an interface in which they are published.

- The author of a module needs to take into consideration the possibility that a public element may be referred to from another module, particularly when

making changes to the module. Where the scope or context of a goal is changed it may be appropriate to change the goal's identifier to avoid confusion.

2:9.6.3 The module interface provides for the detail of the argument to be hidden to support abstraction of higher-level arguments and for protection of privacy e.g. to protect commercially sensitive information. This is achieved by defining an interface that contains only those elements that are relevant to the related module.

- In Figure 2:9-4 it can be inferred that the hwSafReqts and swSafReqts interfaces each contain a subset of goals, with associated context, that address the support for safety requirements allocated to hardware and software respectively. These interfaces can be published separately to the developer of the 'Hardware' and 'Software' modules without either needing visibility of the full interface of the 'System' module.

2:9.6.4 The module interface enables elements from an interface of a contained module to be visible to those interfacing with the containing module. Provision is made to provide an alias as the identity of the element in the contained module interface. This can be used to support abstraction, ensure uniqueness of element identifiers within the interface and avoid unintended exposure of structural information.

- In Figure 2:9-4 for the 'Software' module, it can be inferred that swSafReqts interfaces includes goals that require support from within the module. These may be supported by an argument captured in the 'Software' module's argument view, or could be supported by the interface from say the 'Applications' module contained within the Software module.
- A goal element (e.g. G:swSafReq1Satisfaction) within the 'Applications' module could be made visible in the 'Applications' module interface. This can be made visible to the 'System' module by publishing within the 'Software' module safReqts interface the [Applications]G:swSafReq1Satisfaction element.
- If the 'Applications' module contained a number of application modules, each relating to a separate application, each module may publish a goal identified as G:swSafReqSatisfaction in its interface. In this case the element identified as G:swSafReqSatisfaction in 'ApplicationX' may be published in the 'Applications' module interface as G:swSafReq1Satisfaction. This is an alias of [ApplicationX] G:swSafReqSatisfaction.

2:9.6.5 There are several methods for identifying elements that require support through an interface. Contextual elements that are relevant to a goal requiring support need to be made public where they need to be considered for compatibility of matching goals in a contract:

1. A goal may be shown in the argument view with the ‘to be supported by contract’ decorator.
2. A goal may be linked to a module reference using a ‘supportedBy’ relationship
3. A goal may be linked to a contract module reference using a ‘supportedBy’ relationship.
4. An away-goal may be included.

2:9.6.6 In cases 1 to 3 above, the goal and relevant context will appear in the module’s interface and will be related to another module through a contract. Where the support is provided by a contract that does not have direct visibility of the module interface in which the requirement for support is declared, the goal to be supported needs to be published in the containing module.

- In Figure 2:9-4 for a goal declared as requiring support in the ‘Safety Requirements’ module interface would need to be published in the ‘System’ module interface such that it is visible to the contract to the relevant supporting module (‘Hardware’ or ‘Software’).

2:9.6.7 In case 4, an away-goal reference can only be made to a goal that exists in an interface that is directly visible to the module in which it is declared.

- In Figure 2:9-4, this means that an away goal cannot be included in the ‘Safety Requirements’ module to reference a goal in the ‘Applications’ module.
- However, it would be possible to create an away goal reference from an argument in the ‘System’ module to a goal in the ‘Applications’ module where that goal has been published in the ‘Software’ module. In this case the module identifier cited in the away goal should be ‘Software’ rather than ‘Applications’ since that is the interface that is visible.

2:9.6.8 Within a module, there is no separate interface for the argument depicted within the architecture view of a module, only of the module in which the argument is contained. Therefore, there is no possibility of creating a contract module between the argument depicted in the argument view and interfaces of modules depicted in the

architecture view. In this case an away-goal reference to a goal declared in the interface of a contained module is the only option available.

- In Figure 2:9-3, this is illustrated by the away goal in the argument view to ContextJust goal in the Confidence module.

2:9.7 Configuration Management of Modular Arguments

2:9.7.1 An appropriate configuration management arrangement for the modules in an assurance case argument is a key enabler to practical reusability and change management. It may be desirable to publish individual modules together with their interfaces in a separate, stand-alone document or file with a unique configuration identifier and version control. Traditional configuration management techniques may require that all documents that cooperate to provide an overall argument hierarchically reference supporting modules at a specific issue version. This would result in a situation where, if one document version is changed, it will necessitate the revision of all of the other linked document, which may entirely circumvent the intent to reduce the impact of change. As modules are able to provide mutual support to each other, this may also create a recursive propagation of changes. In such cases an overarching configuration document that indexes the relevant versions of modules to provide a baseline should be considered. This can be linked to a product baseline which the argument baseline assures.

2:10 Confidence Argument Guidance

2:10.1 Introductory

2:10.1.1 The confidence argument extensions to GSN support the realisation of Risk, Confidence, Conformance arguments as discussed in [8]. The concepts behind the approach and the use of the notation are discussed in further detail in [8], with accompanying examples.

2:10.2 Hypothetical Insulin Pump Example (extract)

2:10.2.1 The following extracts are drawn from those examples to illustrate the approach. For further details on the example see [8]. The example illustrates the application of confidence arguments through consideration of the safety case for a hypothetical insulin pump. The example has been simplified for clarity and is provided for illustrative purposes only. Figure 2:10-1 shows the high-level structure of the risk

argument represented in GSN with associated ACPs where a confidence argument is required.

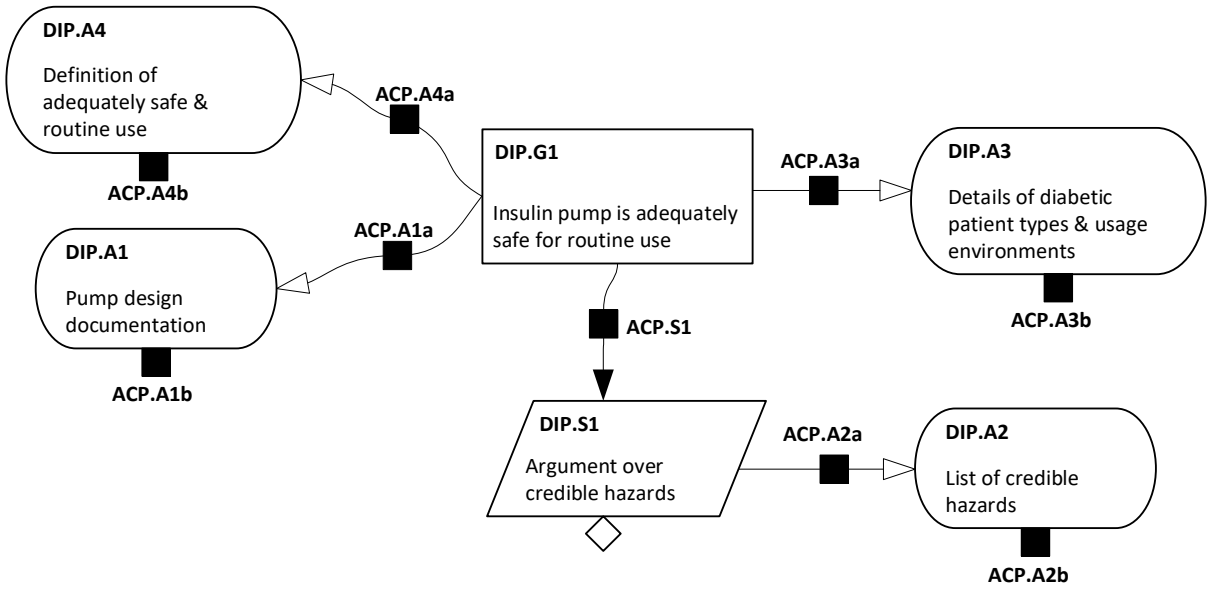


Figure 2:10-1 High-level risk argument for an insulin pump

2:10.2.2 A number of ACPs requiring a confidence argument are identified; this extract focuses on ACP.A1a and ACP.A1b, addressing the confidence in the appropriateness and trustworthiness of the Pump design documentation referenced by DIP.A1 context.

2:10.2.3 Trustworthiness is a property of the artefact referenced by the context element and therefore the ACP.A1b is attached to the context element in Figure 2:10-1. The confidence argument shown in Figure 2:10-2 addresses the trustworthiness aspect. It presents the assurance deficits associated with trustworthiness and addresses each in the confidence argument fragment. For each deficit, counter evidence and sensitivity are considered. Figure 2:10-2 illustrates a claim for counter evidence and sensitivity for claim CC1.3.1. A lack of counter evidence about the commercial word processor based on reported deficiencies is claimed. A lack of sensitivity based on independent information about the design that will be generated by testing and analysis of the pump as built is also claimed. Sensitivity is argued to be low because a defect in the document should be revealed from observations of the pump during testing and analysis.

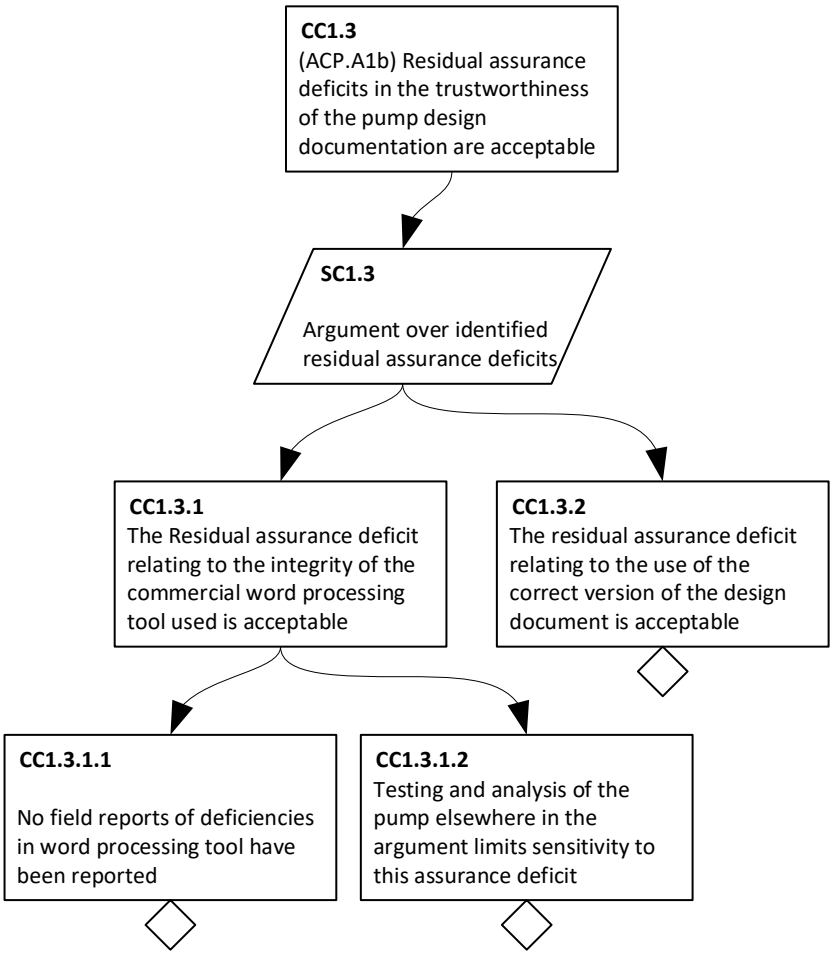


Figure 2:10-2 Trustworthiness Confidence Argument for Context (Partial)

2:10.2.4 Appropriateness is a property of the relationship between the context element and the goal it provides context for, and therefore the ACP.A1a is attached to the 'InContextOf' relationship in Figure 2:10-1. An assurance deficit for the appropriateness sub-argument is shown in Figure 2:10-3 (claim CC2.3.1). For the claim, the problem is that the documentation might be inappropriate because the pump has been locally modified. For lack of counter evidence in this claim, the claim that there is no evidence that such tampering occurs is cited. For sensitivity, the claim that consistency and reasonableness checks by the pump during operation would reveal tampering with a high probability and would raise an alarm is cited. Thus, the remainder of the safety argument is not especially sensitive to this possibility.

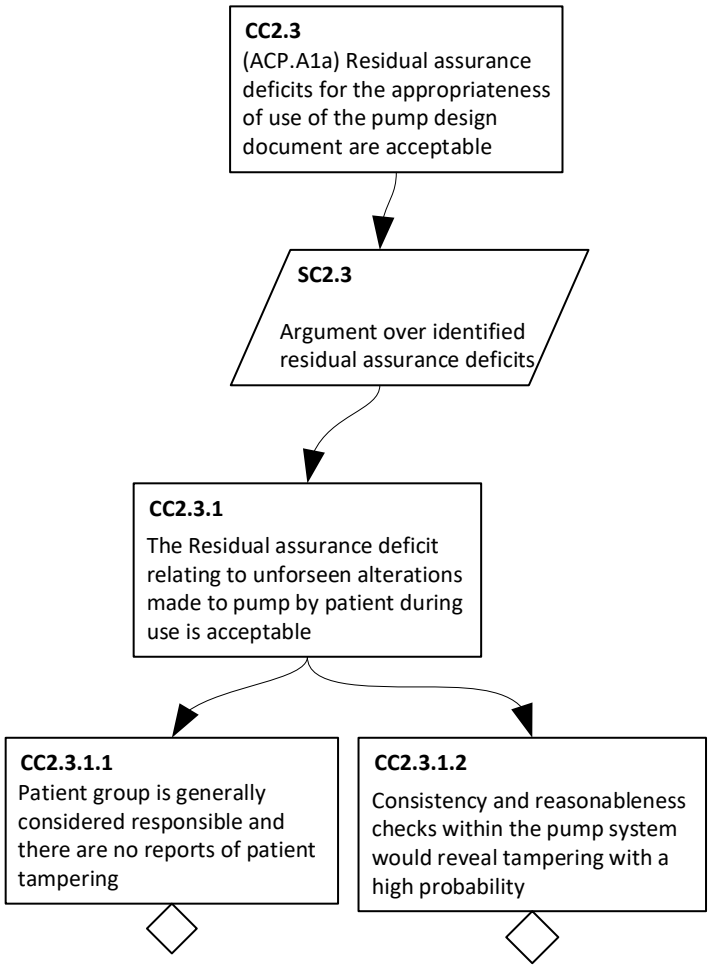


Figure 2:10-3 Appropriateness Confidence Argument for Context (Partial)

2:11 Dialectic Extension Guidance

2:11.1 Introductory

2:11.1.1 This section provides guidance on the use of the dialectic extension to GSN. The Assurance Case Guidance document [8] provides generic guidance on the use of dialectic argument in assurance cases.

2:11.1.2 The term dialectic is used to describe the process of investigating truth. This can occur in a minimal form by simply challenging statements when constructing a goal structure, but can also take a graphical form within a GSN argument. When creating any GSN argument it is recommended that dialectic reasoning is applied. The simplest form of dialectics is to self-question as the goal structure is architected. This can be done initially without necessarily representing the thinking in the goal structure using the dialectic extension. The dialectic extension can then be used as required to depict the dialectic reasoning within the goal structure.

2:11.2 Dialectic Principles

2:11.2.1 This section describes dialectic principles that are commonly applied to dialectic reasoning.

2:11.2.2 Dialectic can be used as a prefix:

- ‘Dialectic argument’ - the outcome of using dialectic thinking or process
- ‘Dialectic element’ - the source of challenge being applied[6]

2:11.2.3 A dialectic argument is based on a set of ‘moves’ or ‘responses’ to any part of the original argument. Such challenges to the argument are achieved by applying ‘defeaters’, which can be directed at any part of an argument. These defeaters are applied in the following way throughout this Guidance[7]:

- As a Rebuttal: interpreted as a supported argument that challenges parts or all of an argument;
- By Undercutting: interpreted as additional facts that are introduced to challenge parts or all of an argument.

2:11.2.4 When applied to a goal structure:

- A GSN Goal can be used to assert a challenge that ‘rebuts’. The rebuttal is complete only once an argument to support the assertion is developed and evidenced, and so a counter-argument is formed.
- A GSN Solution can be used to assert a challenge by presenting a reference to an evidence item that ‘undercuts’. In this way undercutting is documented by identifying counter- evidence.

2:11.2.5 A GSN Challenges relationship declares an inferential or evidential challenge. The inferential relationship declares that there is an inferenced challenge to an element or relationship in the argument (target), asserted by a claim (source element) that is represented by a Goal. The evidential relationship declares there is direct

[6] A dialectic element can sometimes be referred to as a ‘defeater’, though it does not necessarily result in defeat.
[7] It is acknowledged that the use of these terms here and in Section 2:7.4.5 are different and this will be resolved at the next Version.

evidence that challenges an element or relationship in the argument (target) that is referenced by a Solution (source element).

2:11.3 Illustrative Dialectic Example

2:11.3.1 This section describes a staged approach to the development of dialectic reasoning using GSN within goal structures that covers:

- Making dialectic challenges;
- Countering dialectic challenges;
- Applying defeat;
- Propagating defeat.

2:11.3.2 It should be noted that this illustrative example is based on an assumed dialectic process. This is not intended to be definitive and alternative processes may be applied.

2:11.3.3 Using this example process, once a dialectic challenge is made and depicted in a goal structure, this defeats the target of the challenge. Should the original challenge be subsequently countered and itself defeated then the target of the challenge would no longer be depicted as defeated. Alternatively, within another process, it could be consider that once depicted, a challenge needs subsequent steps to consider and resolve the doubt raised.

2:11.3.4 The representation of the resultant dialectic argument may be recorded in one or multiple updates to the GSN structure and some or all of the interim states (dialectic moves) may be transient. The GSN may record various staged outcomes or only the final outcome.

2:11.3.1 Making Dialectic Challenges

2:11.3.1.1 Figure 2:11-1 depicts an entire goal structure to which four dialectic challenges are applied and so directly defeat part of the argument within the goal structure.

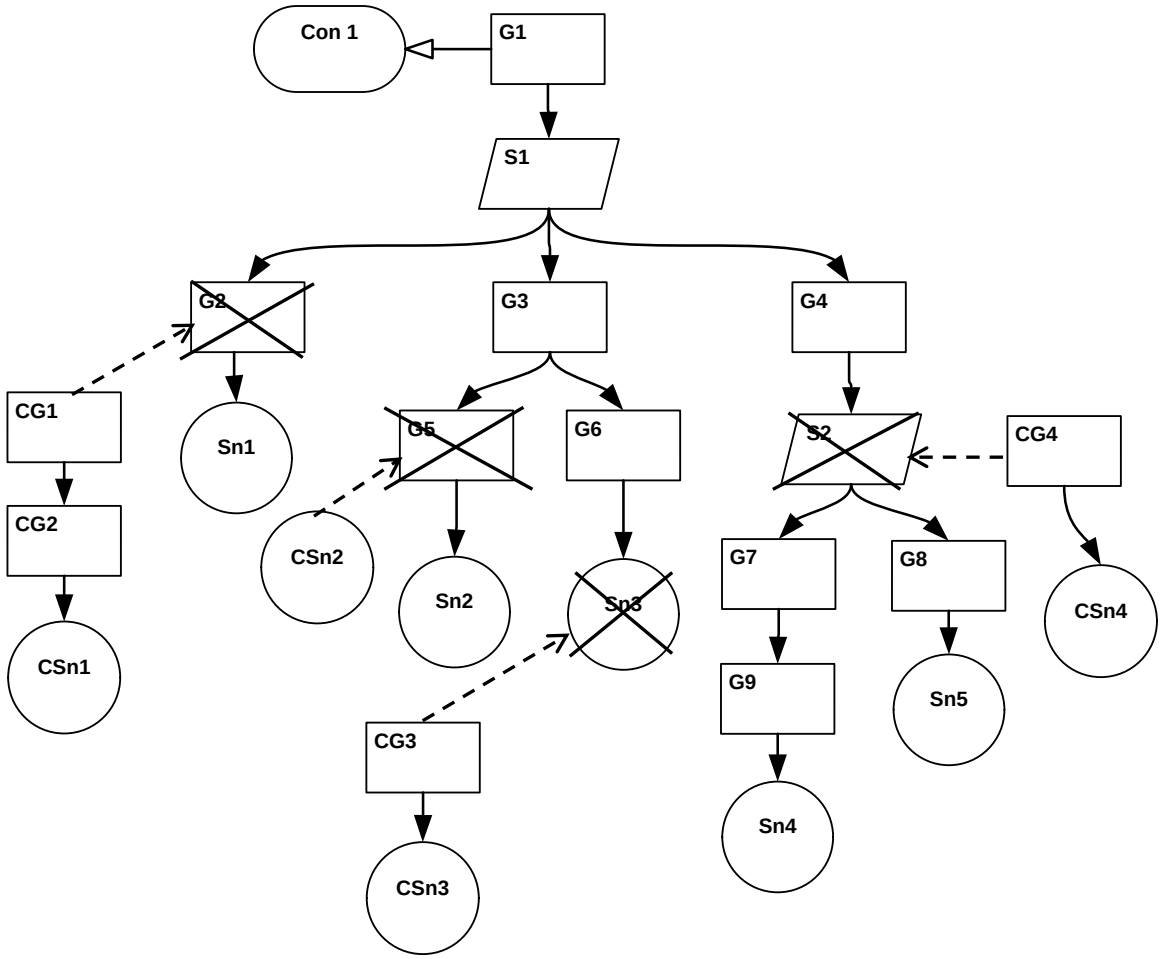


Figure 2:11-1 Application of Dialectic Challenge to a Goal Structure

2:11.3.1.2 A defeater (goal or solution) can challenge any element in a goal structure, e.g. goal, solution, strategy, context, assumption, justification. In this example, two goals and a solution in the goal structure have dialectic challenges applied:

1. The claim presented in goal CG1, as supported by the depicted evidenced argument, rebuts and so defeats the claim presented in goal G2. The defeat is depicted by the defeated decorator to indicate that goal G2 is no longer valid and so presents a claim left as defeated in the goal structure.
2. The evidence referred to in solution CSn2 undercuts and so defeats the claim presented in goal G5. The defeat is depicted by the defeated decorator, which is applied to goal G5 similarly to the defeat in 1 above.
3. The claim presented in goal CG3, as supported by the depicted evidenced argument, rebuts and so defeats the evidence referred to in solution Sn3. The defeat is depicted by the defeated decorator to indicate that Solution Sn3 is no longer valid and so presents evidence left as defeated in the goal structure.

2:11.3.1.3 A defeater (goal or solution) can challenge any relationship in a goal structure i.e. SupportedBy, InContextOf, Challenges as illustrated by this example.

2:11.3.1.4 As the inference between a goal and multiple supporting goals is indivisible the inference can only be challenged in its entirety, so challenges to the SupportedBy relationship is subject to limitations. If the supporting goals are not necessarily sufficient and suitable then such a challenge can only be made directly on a unitary SupportedBy relationship. If the inference is represented by multiple SupportedBy relationships then a strategy must be present or introduced to allow a challenge to be made to the inference, by challenging the strategy. It may also be more intuitive to add and challenge a strategy even for a unitary supporting-goal, although the challenge can be made without such an addition. For a unitary supporting goal, a judgement can be made as it is often desirable to challenge the argument as written and so without update to apply dialectics.

2:11.3.1.5 Two dialectic challenges on inference relationships in the goal structure are considered:

1. The inference asserted by the SupportedBy relationships between goals G4 and its supporting goals, G7 and G8, cannot be challenged directly in the absence of a strategy to represent this inference.
2. The claim presented in goal CG4, as supported by the depicted evidenced argument, rebuts and so defeats the inference which is asserted as existing between goals G4 and its supporting goals, as described by strategy S2. The defeat is depicted by the defeated decorator to indicate that Strategy S2 is no longer valid.

2:11.3.2 Countering dialectic challenges:

2:11.3.2.1 Challenging parts of the goal structure does not necessarily mean that those parts are ultimately defeated. Any challenge within a goal structure can itself be counter challenged, so restoring the target of the original challenge to a valid status.

2:11.3.2.2 Figure 2:11-2 begins with the goal structure shown in Figure 2:11-1, which includes dialectic challenges. Some of these dialectic challenges are countered and subject to further challenges to bring about their defeat.

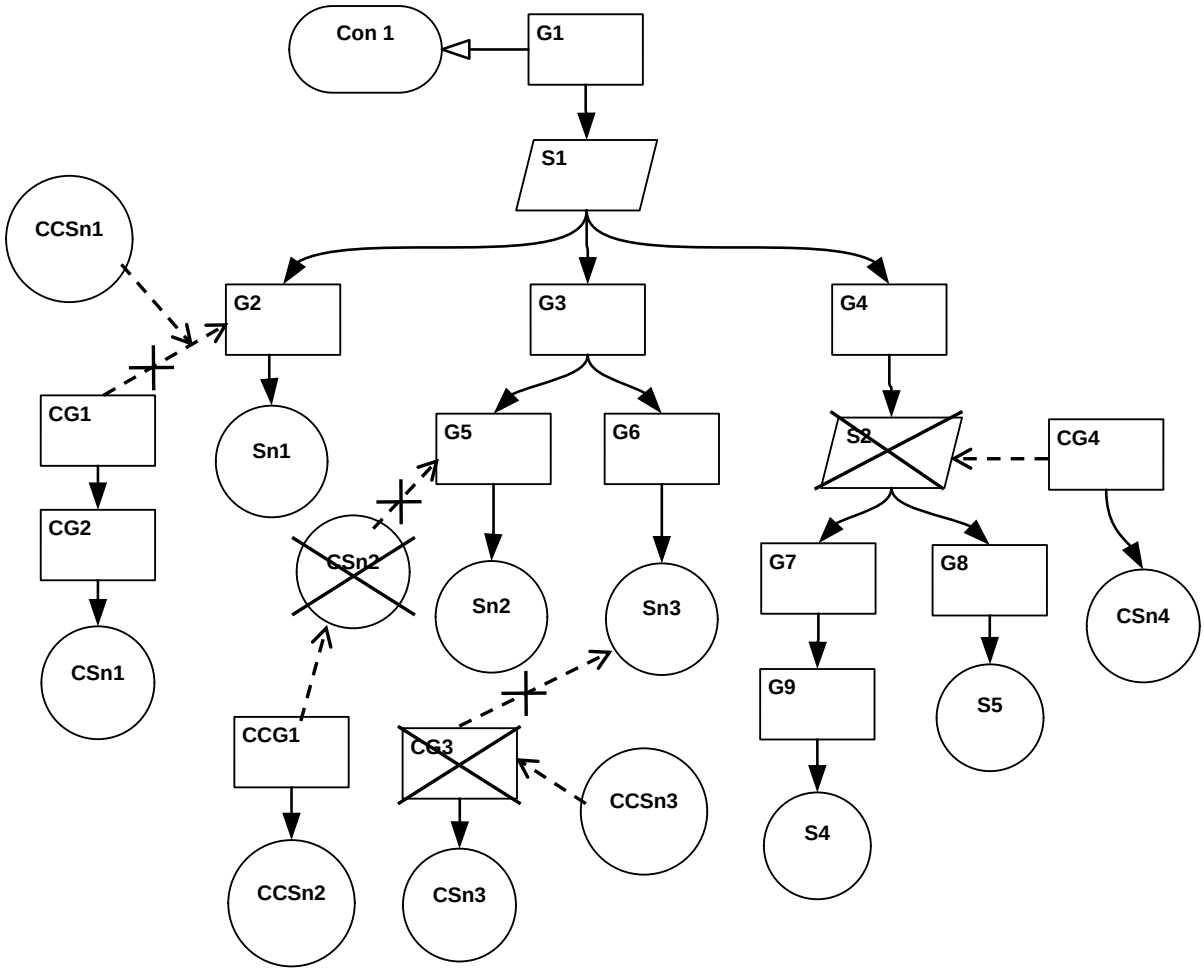


Figure 2:11-2 Countering of Dialectic Challenges

2:11.3.2.3 Counter-challenges are applied to three of the previous challenges depicted in Figure 2:11-1:

1. The claim referred to in goal CCG1 as supported by the depicted evidenced argument, rebuts and so defeats the evidence presented originally in solution CSn2 that would otherwise have undercut goal G5. The defeat is depicted by the defeated decorator, which is applied to indicate that Solution CSn2 is no longer valid and so presents evidence left as defeated in the goal structure. The defeat of the original evidential challenge on goal G5 is depicted by the defeated property applied to the Challenges relationship indicating that the relationship is defeated and no longer valid and so presents this Challenges relationship left as defeated in the goal structure. Having successfully countered the challenge on goal G5, this goal is now no longer depicted as defeated and so presents a valid claim.

2. The evidence referred to in solution CCSn3 undercuts and so defeats the claim presented originally in goal CG3 that would otherwise have rebutted solution Sn3. The defeat is depicted by the defeated decorator, which is applied to indicate that goal CG3 is no longer valid and so presents a claim left as defeated in the goal structure. The defeat of the original referential challenge on solution Sn3 is depicted by the defeated property applied to the Challenges relationship indicating that the relationship is defeated and no longer valid and so presents this Challenges relationship left as defeated in the goal structure. Having successfully countered the challenge on solution Sn3, this solution is now no longer depicted as defeated and so presents a reference to a valid evidence item.
3. The evidence referred to in solution CCSn1 undercuts and so defeats the inference of the original challenge represented by the Challenges relationship sourced from the defeater goal CG1. The original inferential challenge would otherwise have rebutted goal G2. The defeat of the original challenge is depicted by the defeated property applied to the Challenges relationship indicating that the relationship is defeated and no longer valid and so presents this Challenges relationship left as defeated in the goal structure. Having successfully countered the challenge on goal G2, this goal is now no longer depicted as defeated and so presents a valid claim.

2:11.3.3 Successful challenge resulting in defeat

2:11.3.3.1 The claim presented in goal CG4, as supported by the depicted evidenced argument is considered to be valid and is not countered by a further challenge. The defeat depicted by the defeated decorator applied to strategy S2 indicates that strategy S2 remains defeated. Thus, strategy S2 is left as defeated in the goal structure and the inference that strategy S2 describes between goal G4 and its supporting goals G7 and G8 remains invalid.

2:11.3.4 Propagating defeat:

2:11.3.4.1 Defeat identified as a result of dialectic challenge can be propagated throughout the goal structure. Whether to propagate the defeat is a matter for expert judgement and the more likely outcome is that the original goal structure is refactored at this point, particularly where the challenges concerned are viewed as strong and

valid. Once the goal structure is updated in-line with the other applicable normative parts of this standard, further dialectic challenge can be applied until the goal structure is judged to be satisfactory (see Section 2:11.3.4.8).

2:11.3.4.2 Figure 2:11-3 begins with the goal structure shown in Figure 2:11-2 that includes dialectic challenges some of which are countered. The defeat that remains is propagated through the goal structure.

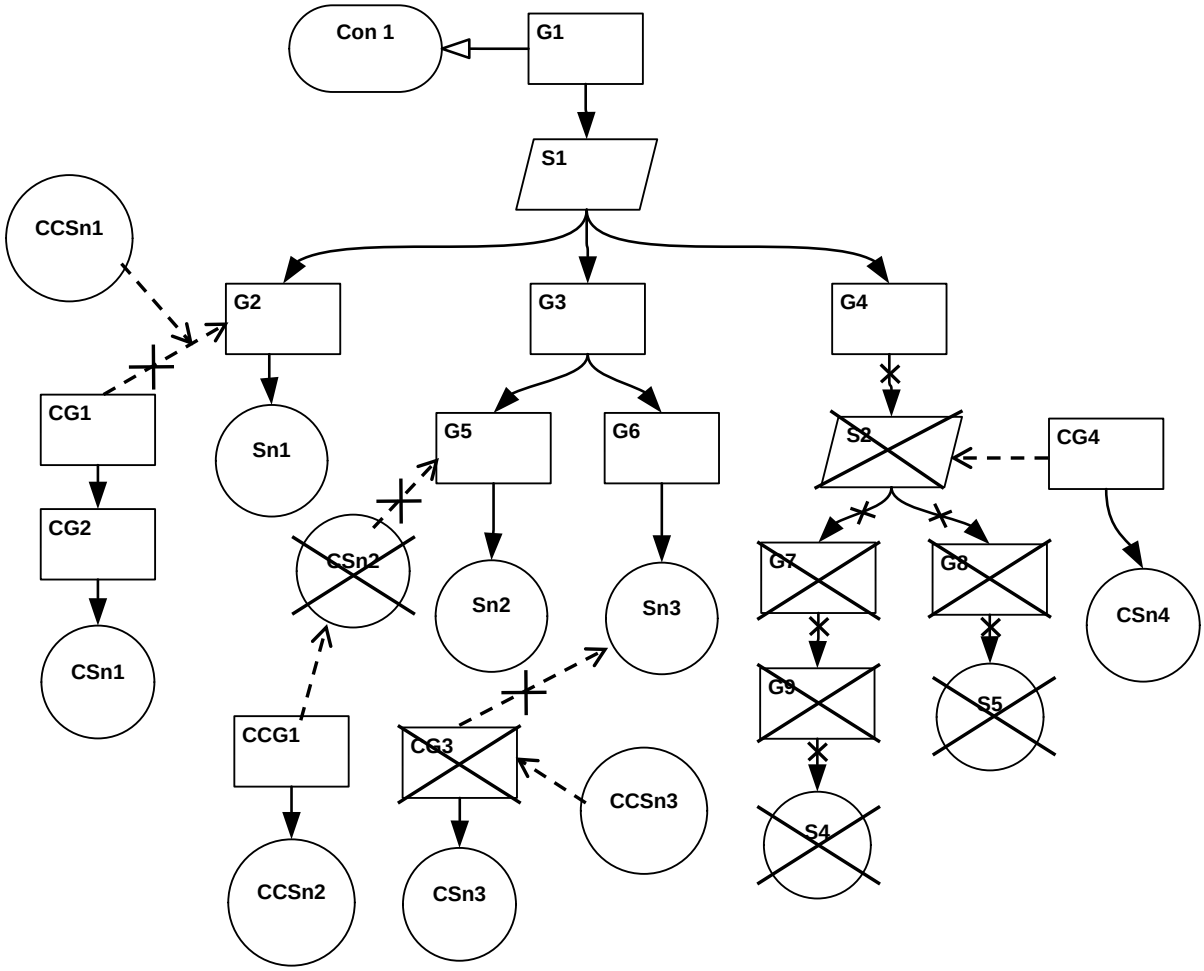


Figure 2:11-3 Propagating Defeat

2:11.3.4.3 Propagation of defeat is case specific and judgement is required to establish the extent of the impact of successful defeat on the surrounding parts of the goal structure. The outcome of propagation is dependent on the case specifics and the judgement applied so the results may vary.

2:11.3.4.4 Expert judgement is used to provide a case specific propagation of the defeat of strategy S2:

1. The inference that strategy S2 describes between goal G4 and its supporting goals G7 and G8 is invalid. In this case it is considered that goals G7 and G8, as supported by the depicted evidenced argument, are not sufficient and suitable to support goal G4. The defeat in this case has resulted in review further down the goal structure as depicted in Figure 2:11-3.
2. The claim presented in goal G4 is considered to be true, thus goal G4 continues to be depicted as undefeated and so presents a valid claim. Goal G4 is now unsupported and so this would need to be resolved in the final goal structure.

2:11.3.4.5 Figure 2:11-4 depicts a goal structure fragment to which a dialectic challenge is applied. The challenge is not countered, so the defeat remains and it is propagated through the goal structure.

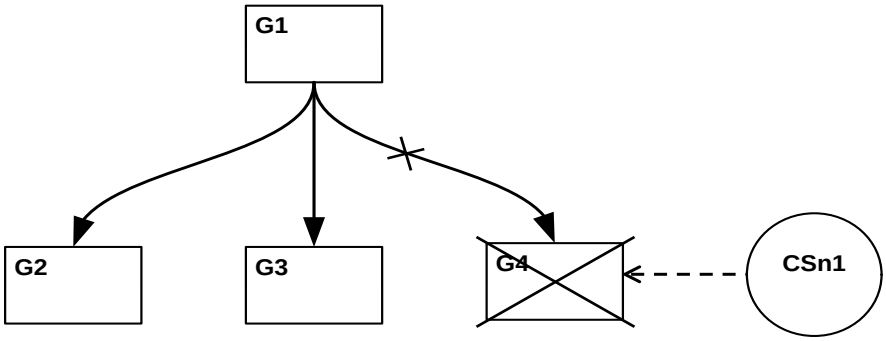


Figure 2:11-4 Propagating Defeat

2:11.3.4.6 The evidence referred to in solution CSn1 undercuts and so defeats the claim presented in goal G4. The defeat is depicted by the defeated decorator, which is applied to indicate that goal G4 is no longer valid and so presents a claim left as defeated in the goal structure. Expert judgement is used to propagate the defeat and so the impact on the inference between goal G1 and all its supporting goals:

1. As the inference between a goal and its supporting goal(s) is indivisible (see Section 1:6.3.11), when propagating defeat in the goal structure the impact on this entity depends on the details of the argument and it may or may not result in full defeat of the supported goal.
2. In this case goal G1 is determined to be still valid, when supported by the remaining supporting goals, G2 and G3, although the strength of support for the claim of goal G1 is weakened. To assist with understanding of the rationale that might apply in such a circumstance, goals G1 and G2 could, say, address product and process argument respectively, whilst goal G3 could address field

evidence. Goals G1 and G2 could still support a higher level goal, but the absence of goal G3 weakens the argument. The argument within any other part of the goal structure that relies on goal G1 for support is also weakened. Consequently, the overall argument depicted by the goal structure is weakened by the defeat of one goal / leg. In this case the remaining argument is still valid.

2:11.3.4.7 Where any part of a challenge is defeated by further challenges, this defeat also needs to be propagated through the challenge to determine the impact on the goal structure that would otherwise be rebutted or undercut.

2:11.4 Presenting Dialectic Argument in Goal Structures

2:11.4.1 GSN 'Challenges' relationship should clearly connect the source of the challenge with its target. No other restrictions on connectivity are applied regarding connection points as this level of flexibility is required to accommodate the dialectic extension, whilst avoiding unnecessary change to the original goal structure.

2:11.4.2 Dialectics may be used during the development of a goal structure and may exist also in a goal structure that is considered complete. Defeaters that remain in a completed version of the goal structure may represent residual doubts that have been accepted. Dialectics may also exist in a completed version of the goal structure to show the use of the dialectic process, even where challenges have been successfully resolved.

2:11.4.3 The dialectic extension should be used to support the following activities, as applicable:

1. Dialectics in forming an acceptable argument (by Assurance Case practitioners for dialectic authoring and on-going review).
2. Evidence of application of dialectics in reaching an acceptable argument (audit trail).
3. Dialectics in the goal structure that shows the dialectic challenges and their resolution or retention (presenting the final argument).
4. An additional goal structure can be created to support and document the review process (providing a formal record of a review activity, for example pertaining to auditors, customers, users / operators, regulators, investigators, etc.).

2:11.4.4 If an element or relationship is defeated, how this propagates throughout the goal structure must be considered. The extent to which defeat is propagated within

the goal structure is case specific and a matter of expert judgement, noting that defeat of an element or relationship does not necessarily mean that the whole goal structure is defeated. It is common practice for a goal structure to be accompanied by narrative. The rationale for the decision making for the propagation of a defeat should be recorded and it is suggested that this be included in accompanying narrative.

2:11.4.5 Should the remaining defeat of one or more parts of the goal structure have a wider impact on the validity of the goal structure then revaluation is required for the final version. Where the dialectic aspects invalidate all or part of the goal structure, then the goal structure should be updated to address the invalidities or the shortfall should be acknowledged in another way (e.g. additional mitigations and controls).

2:11.4.6 There are choices to be made by the practitioner on how to present dialectic argument in the final goal structure. Dialectic argument can be retained and feature in a final goal structure to add strength or it can be preserved elsewhere by other means. It is suggested that this could be achieved for example, through version control, the creation of a side-argument using the modular extension or maybe graphical layers could be used for visibility control.

2:11.4.7 Where dialectic argument results in updates to the goal structure and retention of the original structure with the dialectic challenges is preferred (depicted as defeated), an additional goal can be added beneath the higher goal that has not changed to address the impact of defeat. Various suggested options for managing visualization of retained dialectic argument include:

- tooling that supports the hiding and selection of detail in goal structures;
- use of context to make reference to dialectic argument that has been addressed.

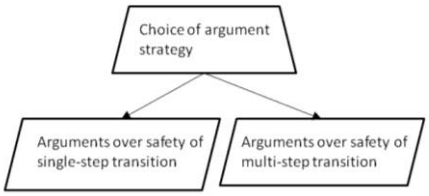
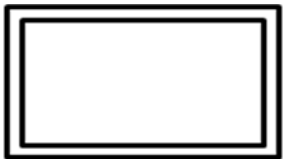
2:11.4.8 A common dilemma concerning the use of dialectic argument centres on: ‘when I have done enough’; ‘when should I stop’. In essence the solution lies in a systematic process that is run to completion and applied using expert judgement. The Assurance Case Guidance document [8] provides more detail.

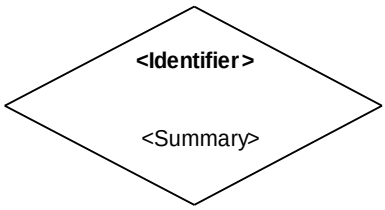
Annex 2:A Deprecated Extensions to GSN

2:A.1 From time to time, elements other than those defined in Part 1: of this Standard may be encountered in goal structures. These have formed part of the notation as it has evolved but are considered by the standard as deprecated. Table 2:11.1 illustrates such elements and explains the concepts they are intended to represent.

2:A.2 With the exception of the choice-of-strategy element, all of these symbols can be replaced by suitably worded context elements without serious loss of meaning. They are therefore considered redundant, and their use is discouraged.

Table 2:11.1 Non-Modular Extensions to GSN

GSN Element Rendering	Definition
Strategy choice 	This structure signifies that there is a choice still to be made about how the argument will be constructed. A choice should never appear in a final argument structure but may be helpful in developing the argument and exploring the implications of alternative possibilities. In the example shown the project has not decided on its strategy for transition to operations. It can be replaced by the solid diamond choice symbol used in GSN patterns.
Criterion 	This is a form of context symbol which is used to indicate a criterion by which the goal to which it is attached will be regarded as appropriately supported. Example: 85% statement test coverage regarded as meeting this goal
Constraint 	This is a form of context symbol which is used to indicate a constraint that might impact the way in which the goal to which it is attached can be supported. Example: Source code of component not available for inspection

GSN Element Rendering	Definition
Stakeholder 	This is a form of context symbol which is used to indicate one of the stakeholders associated in some way with the goal to which it is attached. Example: Installation Contractor (ABC Cabling Ltd)
Problem 	This is a form of context symbol used to indicate a problem associated with the goal to which it is attached, and may be used to indicate that there is counter-evidence which casts doubt on the goal's validity. The use of colour or shading is the only way in which this shape is distinguished from a goal, but a problem can only appear attached to a goal as context. This may be replaced by dialectic argument. Example: In-service trial reported several failures contradicting predictions of FTA.
Model 	This is a context symbol which refers to an information artefact in the form of a model.

Annex 2:B Converting a textual argument into GSN

2:B.1 Where an assurance case has been presented textually, it can be useful to re-express the argument in GSN to be able to visualise the structure and content of the argument. This can be accomplished with the following steps.

2:B.2 Step 1: To identify the essential elements of the assurance case, mark the text with coloured highlights, identifying each element in the argument (evidence, assumptions, claims etc.).

2:B.3 Step 2: Identify the links between them. This activity involves determining the argument approaches which are being used to support the claims identified and the evidence items being used to support the arguments. If these links are not immediately obvious from the text of the assurance case report, it will be necessary to cross-reference the text within the document.

2:B.4 Step 3: At this point, the argument can be re-represented using GSN and evaluated for the key properties of an argument as described in section 2:7.

GLOSSARY

Argument

A body of information presented with the intention to establish one or more claims through the presentation of related supporting claims, evidence and contextual information.

Assurance Case

Arguments and evidence intended to demonstrate that a system meets its assurance requirements.

Claim

A proposition being asserted by the author that is a true or false statement.

Dialectic

The process of investigating truth. This can occur in a minimal form by simply challenging statements made in an assurance case, but can also take a graphical form within a GSN argument

Evidence

Information or objective artefacts being offered in support of one or more claims.

Evidential Relationship

A declared relationship between a claim and an evidence item by which the claim is substantiated.

Inferential Relationship

A declared inference between claims in the argument.

Structured argument

A particular kind of argument where the relationships between the asserted claims, and from the evidence to the claims, are explicitly represented.

Top Goal

A GSN Goal that presents the pinnacle claim in an argument. It is ‘top’ in terms of the argument hierarchy, rather than necessarily its physical layout. There may be more than one top goal in a GSN structure.

REFERENCES

[1] S. Wilson, J. McDermid, P. Fenelon and P. Kirkham, 'No More Spineless safety Cases: A Structured Method and Comprehensive Tool Support for the Production of Safety Cases', presented at the 2nd International Conference on Control and Instrumentation in Nuclear Installations (INEC'95), Cambridge UK, 1995.

[2] S. Toulmin, The Uses of Argument (1958; 2nd edition, 2003).

[3] A. Dardenne, A. van Lamsweerde and S. Fickas, 'Goal-Directed Requirements Acquisition', Science of Computer Programming 20 (1993).

[4] T. Kelly, 'Arguing Safety: A Systematic Approach to Managing Safety Cases', D.Phil Thesis, University of York (1998).

[5] T. Kelly, 'Reviewing Assurance Arguments – A Step-by-Step Approach', in Proceedings of the Workshop on Assurance Cases for Security – The Metrics Challenge, Dependable Systems and Networks (DSN), July 2007.

[6] RTCA/DO-178C Software Considerations in Airborne Systems and Equipment Certification (RTCA, 2011)

[7] Object Management Group (OMG), Structured Assurance Case Metamodel (SACM), URL: http://www.omg.org/spec/SACM.

[8] SCSC-159: Assurance Case Guidance 2021: SCSC Assurance Case Working Group [https://scsc.uk/scsc-159].

GOAL STRUCTURING NOTATION (GSN)

COMMUNITY STANDARD (Version 3)

This Standard has two intended functions. Firstly, it seeks to provide a comprehensive, authoritative definition of the Goal Structuring Notation (GSN). Secondly, it aims to provide clear guidance on current best practice in the use of the notation for those concerned with the development and evaluation of engineering arguments – argument owners, readers, authors and approvers.

The first version of the Standard was developed by means of a consensus process involving GSN users from both academia and industry, between 2007 and 2011. It is thus a standard written for the community, by the community.

The second and third versions reflect comments and suggestions from users based on their experiences using the notation. The document history outlines the recent history of the collaboration, and a list of contributors to the Standard is provided on page 5.

Responsibility for publication and maintenance of this standard has now been transferred to the SCSC Assurance Case Working Group (ACWG). See www.scsc.uk/gc for further details