GSN COMMUNITY STANDARD VERSION 1

November 2011

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 – argument owners, readers, authors and approvers.

The Standard was developed by means of a consensus process involving GSN users from both academia and industry, between 2007 and 2011. The document history on page iii outlines the recent history of the collaboration, and a list of contributors to the Standard is provided on page iv.

DOCUMENT HISTORY

Version	Issued to	Date	Purpose
Draft 0.3	Standardisation Committee	2nd May 2010	For review prior to general circulation
Draft 1.0	User Community and interested parties	19th May 2010	For consultation and comment
Draft 1.1	Standardisation Committee	3rd September 2011	For review prior to issue
Issue 1	User Community	16th November 2911	For use

INDIVIDUAL CONTRIBUTORS

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

CONTRIBUTING ORGANISATIONS

AACE Ltd
Altran Praxis Ltd
Columbus Computing Ltd
ERA Technology Ltd
LR Rail Ltd
RPS Group Ltd
Selex-Galileo Ltd
UK Ministry of Defence
Adelard LLP
BAE Systems Ltd
CSE International Ltd
General Dynamics UK Ltd
Origin Consulting (York) Ltd
SafeEng Ltd
Thales Ltd
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 argumentation notation that can be used to document explicitly the individual elements of any argument (claims, evidence and contextual information) and, perhaps more significantly, 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). Arguments documented using GSN can help provide assurance of critical properties of systems, services and organisations (such as safety or security properties).

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 and how the notation relates to basic principles of argumentation.
- **Part 1: Definition of GSN** (normative)**.** This part is divided into two sections. The first section provides a normative definition of the syntax of GSN, including its visual syntax. In the second section, which is intended to be more informative, the semantics of the notation is provided, clarifying the meanings of standard GSN structures. Annexes to Part 1 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 Use of GSN** (informative**).** This part provides informative guidance on the effective use of GSN to create and evaluate structured arguments.
- **Part 3: Web-Based Resources (still in development).** This part provides additional informative guidance on the use of GSN, including further examples of goal structures and catalogues of existing argument patterns. These resources are currently under development, and will be made available in due course.

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 Arguments presented using GSN can help provide assurance of critical properties of systems, services or organisations (such as safety or security properties). Such arguments can form a key part of an overall assurance case. The role of arguments in assurance cases is explained in Section 0.2.

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.

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 safety-case stakeholders. It should also be clear how the evidence is being asserted to support this argument. By appealing to core concepts of argumentation, GSN helps address these objectives.

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, we make supporting claims. These claims may themselves need further support. This gives rise to a hierarchy of claims (representing a logical chain of reasoning) by which an argument is established. At

the heart of GSN is the explicit documentation of this hierarchy of claims. 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 documented in *solutions*. GSN element names are italicised throughout this Standard, to distinguish reserved uses of those words from ordinary usage. The relationships represented in GSN are:

- The predicate-conclusion relationship between goals and sub-goals;
- The support that solutions provide for claims; 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 (represented in GSN as *goals*) are said to be supported by sub-claims (also represented in GSN as *goals*). Figure 1 shows an example *goal* in GSN:

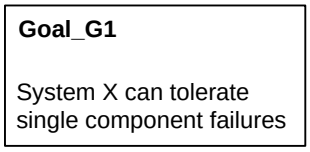


Figure 1: An Example Goal

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 2 shows an example *solution* (reference to evidence) in GSN:

Figure 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 3 shows an example *strategy* in GSN:

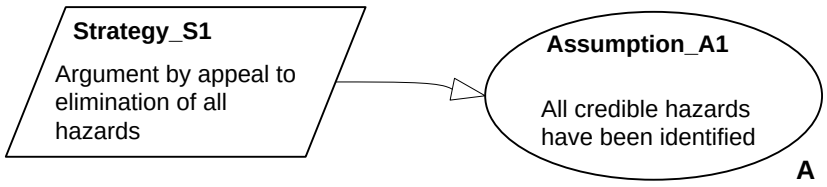


Figure 3: An Example Strategy

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 4 shows an example *context* in GSN:

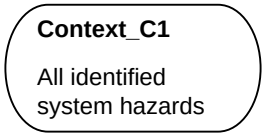


Figure 4: An Example Context

0.4.6 Some claims and argument strategies rely on assumptions to hold valid. These assumptions can be documented explicitly in GSN using the *assumption* element. An example of an *assumption* can be seen in Figure 3: Strategy S1 relies on an assumption that all credible hazards have been identified correctly in order for the line of argument it leads to be persuasive.

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 5: the argument author justifies the use of an argument approach using SILs by asserting that SIL apportionment is recognised by an appropriate safety standard.

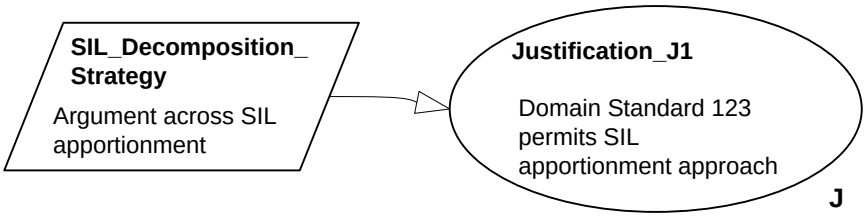


Figure 5: An Example Justification

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.

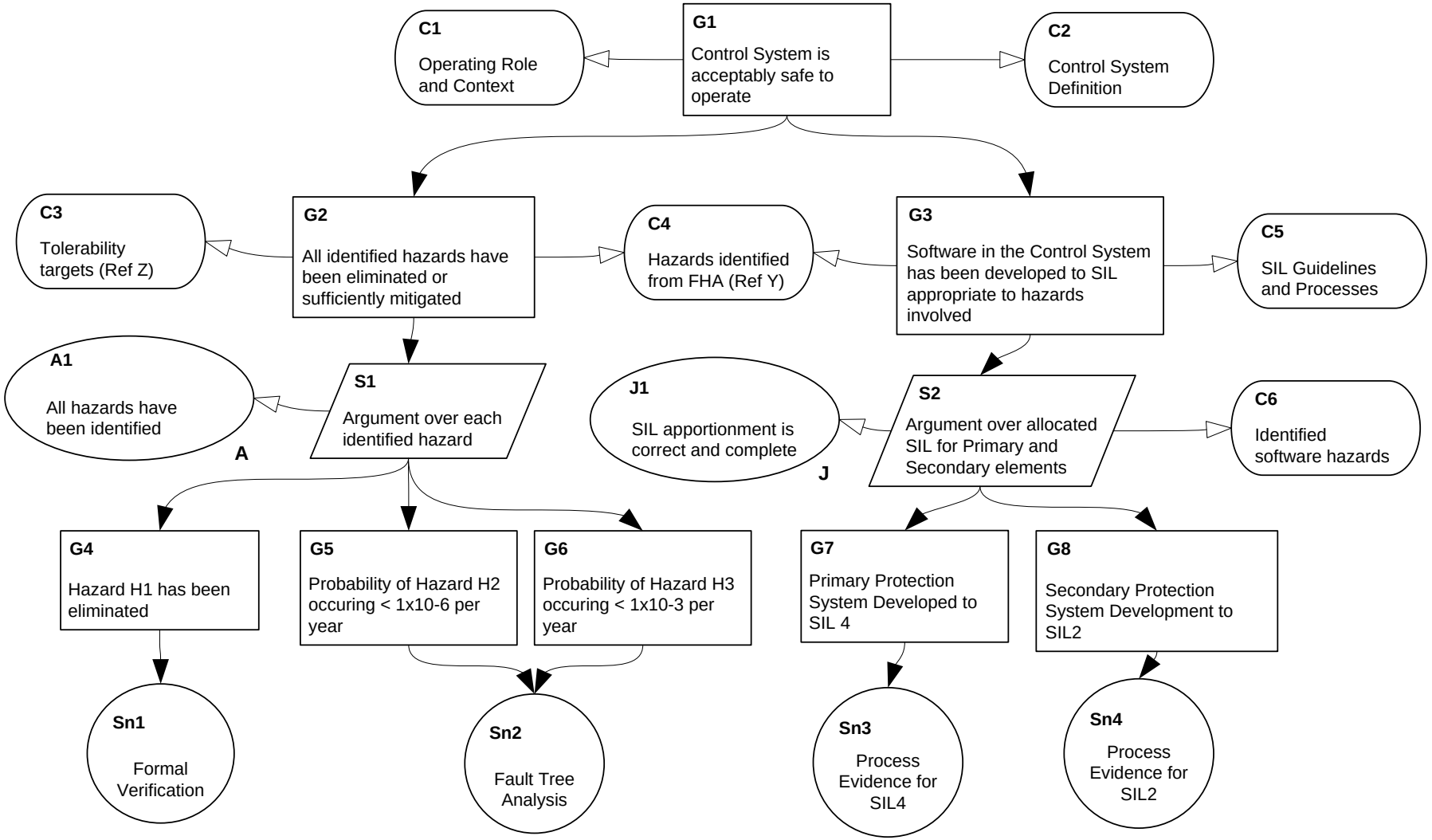


Figure 6: An Example Goal Structure

0.4.10 When the elements of GSN are connected together, they are said to form a ‘goal structure’. Figure 6 shows 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. system developers, engineers, independent assessors and certification authorities). In turn, this improves 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 6, 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. The core elements of the notation are introduced in Section 1.2. Section 1.3 describes the interpretation of permitted combinations of these elements. Section 1.4 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 Annexes A1 and B1.

1.1.2 GSN was originated at the University of York in the early 1990s as part of the ASAM-II project [**2**], and has undergone significant development and refinement since then. The early development of GSN was heavily influenced by Toulmin's work on argumentation [**3**] and emerging goal-based approaches to requirements engineering, such as KAOS [**4**].

1.2 Notation

1.2.1 GSN defines the following elements:

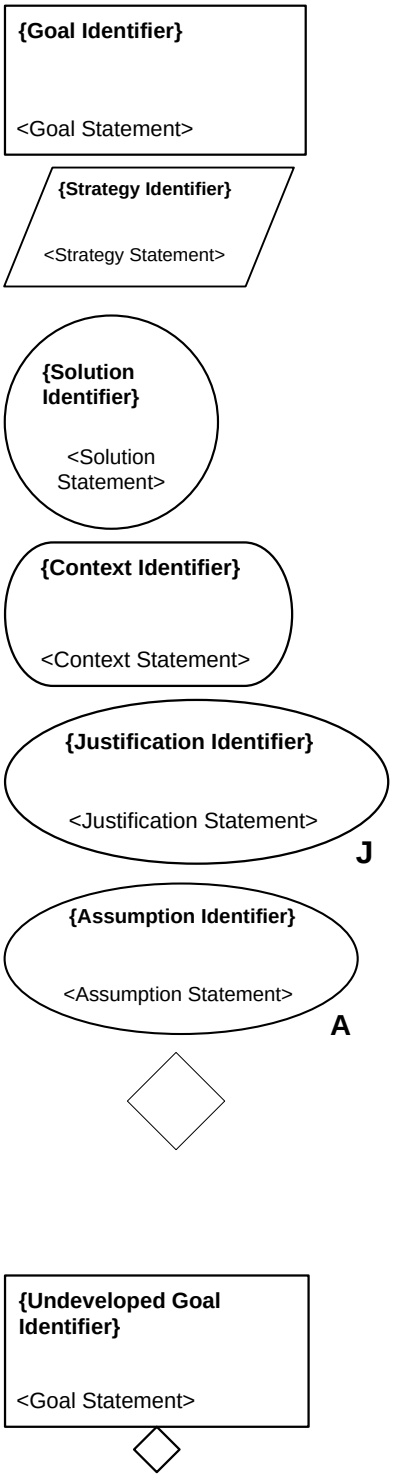
- *Goals*
- *Strategies*
- *Solutions*
- *Contexts*
- *Assumptions*
- *Justifications.*

1.2.2 As indicated below, there is provision for an optional element identifier (represented here by curly brackets). Where it is provided, the identifier should identify the element uniquely.

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

- *SupportedBy*
- *InContextOf*.

1.2.4 Figure 7 provides the definition and rendering of these elements. GSN relationships are defined in Section 1.2.5 below. The meanings of structures combining these relationships are further explained in Section 1.3.



A ***goal***, rendered as a rectangle, presents a claim forming part of the argument.

A ***strategy***, rendered as a parallelogram, describes the nature of 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 or items.

A ***context***, rendered as shown left, presents a contextual artefact. This can be a reference to contextual information, or a statement.

A ***justification***, rendered as an oval with the letter 'J' at the bottom-right, presents a statement of rationale.

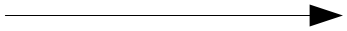
An ***assumption***, rendered as an oval with the letter 'A' at the bottom-right, presents an intentionally unsubstantiated statement.

Undeveloped entity, rendered as a hollow diamond applied to the centre of an element, indicates that a line of argument has not been developed. It can apply to goals (as below) and strategies.

An ***undeveloped goal***, rendered as a rectangle with the hollow-diamond 'undeveloped entity' symbol at the centre-bottom, presents a claim which is intentionally left undeveloped in the argument.

Figure 7: Core GSN Elements

1.2.5 The core GSN elements defined here are intended to be combined to represent logical structures, known as ‘goal structures’. GSN provides two types of linkage between elements, as indicated in Figure 8:



SupportedBy, rendered as a line with a solid arrowhead, allows inferential or evidential relationships to be documented. Inferential relationships declare that there is an inference between goals in the argument. Evidential relationships declare the link between a goal and the evidence used to substantiate it. 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 connections are: goal-to-context, goal-to-assumption, goal-to-justification, strategy-to-context, strategy-to-assumption and strategy-to-justification.

Figure 8: Core GSN Relationships

1.3 Notation Interpretation

1.3.1 Figure 9 shows the most basic relationship represented in goal structures – inference between *goals*:

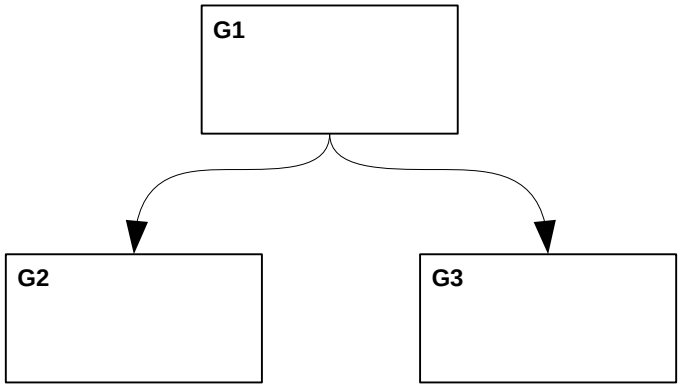


Figure 9: Supporting Goals with Sub-Goals

This specific structure 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. G2 and G3 would

commonly be referred to as ‘sub-goals’, ‘supporting goals’ or ‘child goals’ of G1. This relationship is often referred to as a ‘parent goal – child goal(s)’ relationship. One or more sub-goals may be declared for a given *goal*.

1.3.2 The structure shown in Figure 10 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 nature of the inference which is asserted as existing between sub-goals G2 and G3 and the parent *goal* G1.

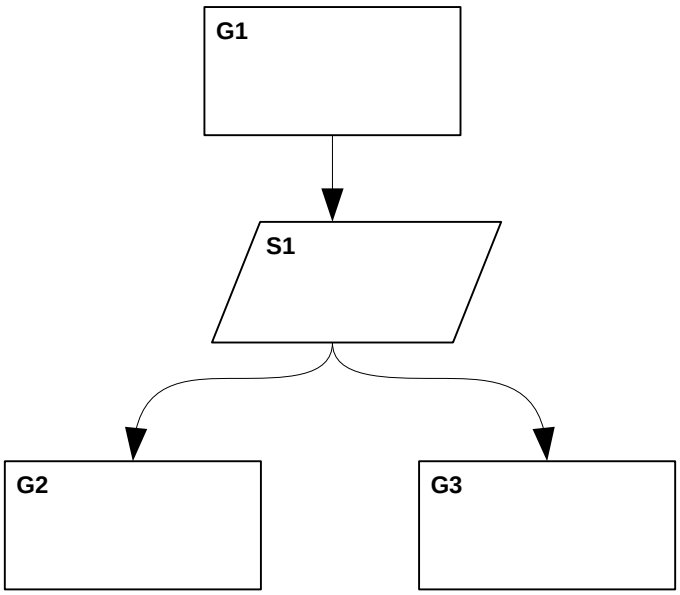


Figure 10: Adding Strategy

1.3.3 In some cases, more than one argument approach may be adopted in support of a parent *goal*. Figure 11 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 sub-goals G2 and G3 to the parent G1. Strategy S2 describes the argument relating G4 and G5 to G1.

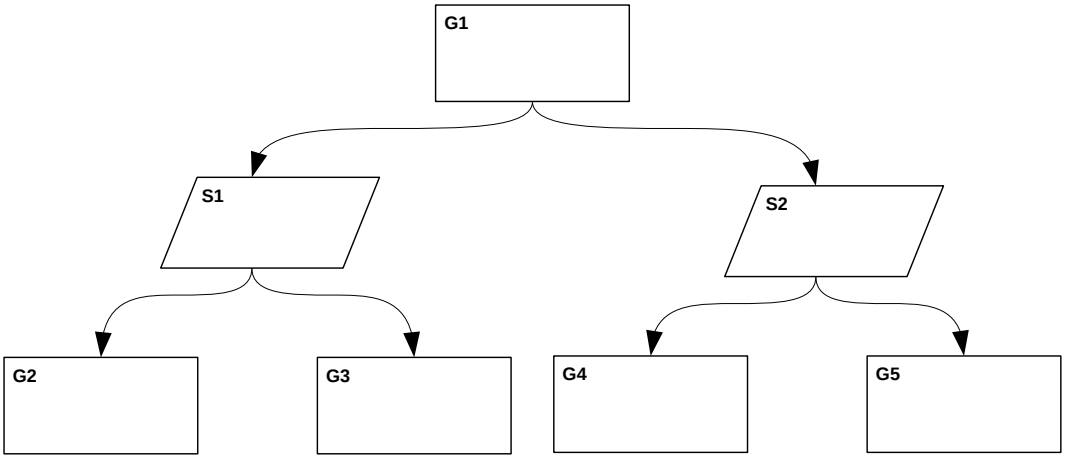


Figure 11: Multiple Strategies

1.3.4 Figure 12 represents the use of a reference to an evidence item to support a claim.

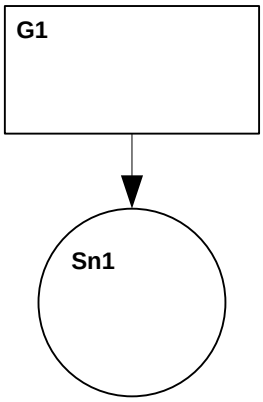


Figure 12: Providing Solutions

This structure represents an assertion that the evidence referred to in the *solution* (Sn1) is sufficient to establish the truth of the claim made in the *goal* (G1).

1.3.5 As with the use of multiple argument approaches to support a claim demonstrated in Figure 11, there may be situations in which the existence of multiple evidence artefacts is invoked in support of a claim. In cases of this kind, multiple GSN *solutions* will be presented in the goal structure. Figure 13 represents an assertion that the evidence referred to in Solutions Sn1 and Sn2 is sufficient to establish the truth of the claim made in Goal G1.

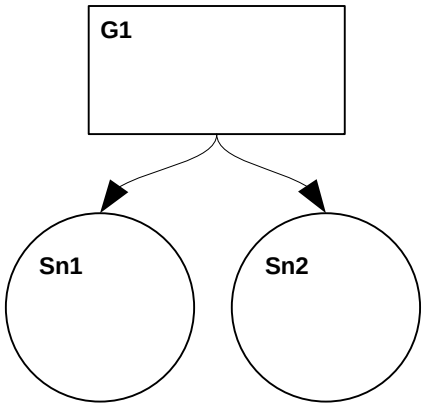


Figure 13: Multiple Solutions

1.3.6 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 14 shows the addition of *context* to a *goal*. The *context* is used to declare supplementary information related to the claim made in Goal G1.

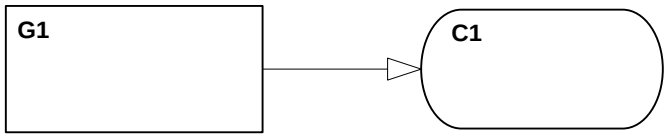


Figure 14: Adding a Context to a Goal

1.3.7 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.*

1.3.8 An *assumption* applied to a *goal* declares an assumption made in stating the claim. The meaning of the structure in Figure 15 is that the claim in Goal G1 is asserted in a context where the assumption in A1 is true:

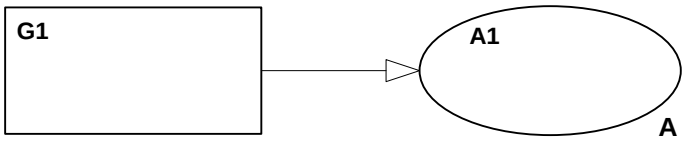


Figure15: Adding an Assumption to a Goal

1.3.9 An *assumption* is an unsubstantiated statement. The scope of an *assumption* is the entire argument. Having connected an *assumption* to a *goal* G1, the assumption is taken to be connected to the entirety of the argument supporting G1. Therefore, it is not necessary to restate the *assumption* in the supporting argument.

1.3.10 Figure 16 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. Should an equivalent justification be required elsewhere in the argument, it will need to be re-stated or re-linked.

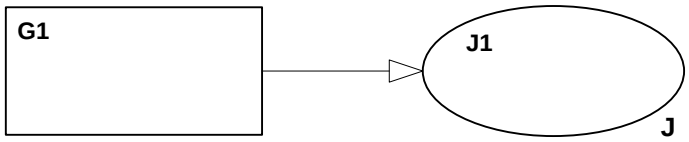


Figure 16: Adding a Justification to a Goal

1.3.11 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 17 shows the addition of a *context* to a *strategy*:

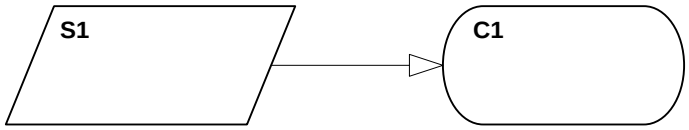


Figure 17: Adding a Context to a Strategy

1.3.12 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*.

1.3.13 An *assumption* applied to a strategy declares an assumption in how the sub-goals support the parent *goal*. In the structure presented in Figure 18, in declaring that the sub-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.

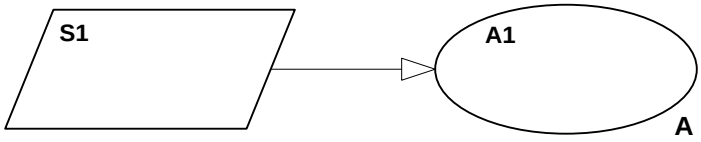


Figure 18: Adding an Assumption to a Strategy

1.3.14 a *justification* can also be connected to a *strategy*, to provide backing for the argument described by the strategy. Figure 19 shows the addition of a *justification* to a GSN *strategy*:

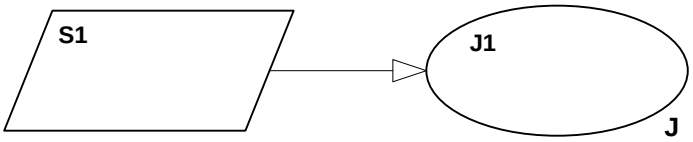


Figure19: Adding a Justification to a Strategy

1.3.15 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.

1.4 The Language of Goal Structures

1.4.1 A series of simple rules governs the grammatical structure of statements used in GSN elements.

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

1.4.3 GSN *strategy* statements describe the reasoning that connects parent *goals* and supporting *goals* in abstract terms, but the core claims and the structure connecting those claims remain unchanged. *Strategy* statements contain a brief description of the argument approach.

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

1.4.5 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.

1.4.6 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.

ANNEXES TO PART 1

A1 EXTENSIONS TO GSN TO SUPPORT ARGUMENT PATTERNS

A1.1 Introductory

A1.1.1 In order to represent patterns of argument rather than merely argument instances, GSN has been extended to support structural and entity abstraction.

A1.1.2 Note that the extensions to core GSN presented in sections A1.2 and A1.3 below are intended for the representation of abstract argument patterns. In cases where the elements defined in these 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.

A1.2 Structural Abstraction in GSN

A1.2.1 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,

A1.2.2 Figure 20 illustrates the extensions made to GSN to facilitate the representation of multiplicity. These symbols are defined for use as annotation on all existing GSN relation types. Multiplicity symbols can be used to describe how many instances of one element-type relate to another element.

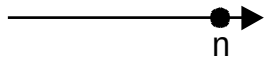
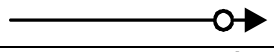
	A solid ball is the symbol for many (meaning zero or more). The label next to the ball indicates the cardinality of the relationship.
	A hollow ball indicates ‘optional’ (meaning zero or one).

Figure 20: GSN Multiplicity Extensions (for Structural Abstraction)

A1.2.3 The extension to GSN shown in Figure 21 enables the representation of structural options using the notation. This figure introduces the GSN *option* symbol, which is rendered as a solid diamond. This symbol is defined for use over all existing GSN relation types. A GSN *option* can be used to denote possible alternatives in satisfying a relationship. It can represent 1-of-n and m-of-n selection, an annotation indicating the nature of the choice to be made. In Figure 21, one *goal* can be supported by any one of three possible sub-goals.

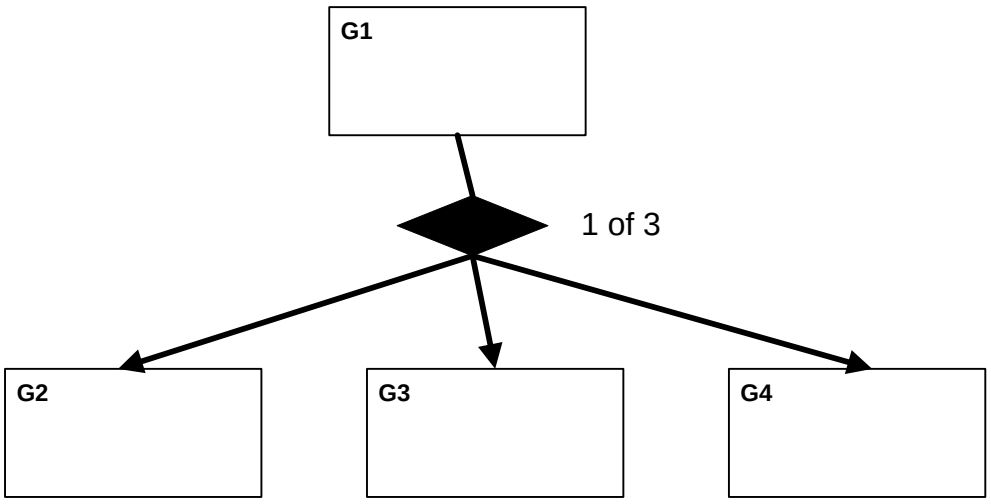


Figure 21: GSN Option Element

A1.2.4 Further guidance on the use of the GSN *option* symbol is provided in Annex A2 below.

A1.3 Entity Abstraction in GSN

A1.3.1 Figure 22 illustrates extensions to GSN to enable the representation of abstract entities:

Uninstantiated Entity	This annotation denotes that the attached entity remains to be instantiated, i.e. at some later stage the 'abstract' entity needs to be **replaced** (instantiated) with a more concrete instance. This annotation can be applied to any GSN element type.
UninstantGoal {Hazard H} has been sufficiently mitigated	Example of an *uninstantiated goal*, demonstrating the application of the annotation.
Undeveloped and Uninstantiated Entity	This annotation denotes that the attached entity requires both further development and instantiation. This annotation can be applied to GSN goals and strategies.
UndevelGoal All hazards have been mitigated	Example of an *undeveloped goal*, demonstrating the application of the annotation

Figure **22**: GSN Extensions for Entity Abstraction

B1 MODULAR EXTENSIONS TO GSN

B1.1 Introductory

B1.1.1 The definition of GSN provided within the main body of Part 1 is typically used for arguments that can be defined in one place as a single artefact rather than as a series of modularised interconnected arguments. This annex describes how GSN has been extended to represent interrelated modules of argument.

B1.2 Notation Extensions

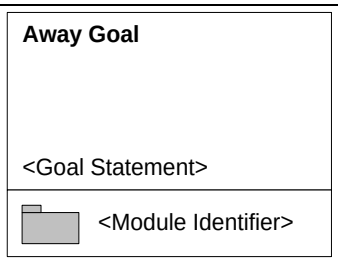
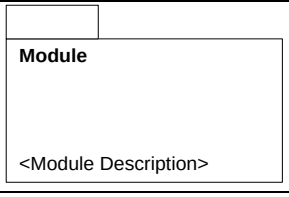
B1.2.1 The following elements are used in addition to the core GSN notation:

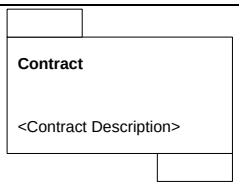
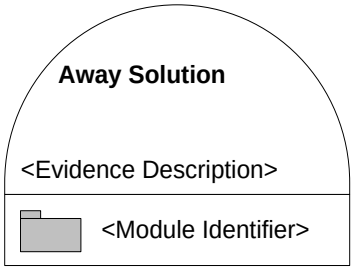
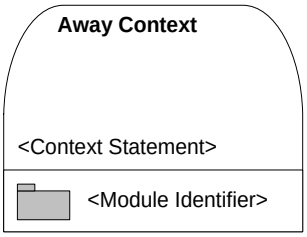
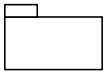
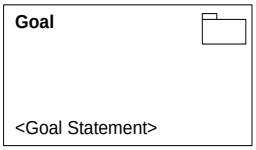
- *Away Goal*
- *Module*
- *Contract*
- *Away Solution*
- *Away Context*.

B1.2.2 The concept of a 'module view' is also introduced. This uses a subset of the extended notation elements to provide an abstract view of the argument structure.

B1.2.3 No new link types are introduced, though the definition of permitted connections and interpretation of the links in the modular view is extended.

B1.2.4 Figures 23, 24 and 25 provide the definition and rendering of these elements and relationships. The meanings of structures combining these elements are further explained in Section B1.3.

	An ***away goal***, 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* symbol. This repeats a claim presented in another argument *module* which is used to support the argument in the local *module*. The Module Identifier provides a reference to the module that presents the original claim.
	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.

	A ***contract module* 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* containing definition of the relationships between two *modules*, defining how a claim in one supports the argument in the other.
	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), repeats a reference to evidence items presented in another argument *module*. The Module Identifier provides a reference to the *module* that presents the original reference.
	An ***away context***, rendered as shown left, repeats a contextual artefact. The Module Identifier provides a reference to the *module* that presents the original reference.
 Public Indicator Symbol  Example of use (goal)	***Public Indicator***, rendered as a miniature *module* symbol and superimposed on a *goal*, *solution* or *context* symbol at the top right. This indicates that the element is publicly visible to other *modules*, and can be referenced as an *away goal*, *away solution* or *away context*.

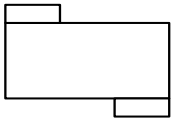
	***To be supported by contract*:** This annotation, attached centrally immediately below the *goal* to which it relates, denotes that support for the claim presented by the attached *goal* is intended to be provided from an argument in another *module*, linked by an as-yet-undisclosed *contract*. At some later stage, the element may be updated to replace this annotation with support from a named *contract module*, or may be left as it is, with the necessary support defined in a higher-level argument abstraction. This annotation 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.

Figure 23: New Entities added to the Core GSN Notation to support Modularity

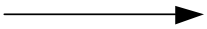
	***SupportedBy*** In addition to the permitted connections defined in the core GSN definition (Section 1.2), in the modular GSN extension the following additional connections are permitted: goal-to-away goal, goal-to-away solution, goal-to-module, goal-to-contract module, strategy-to-away goal, strategy-to-away solution, strategy-to-module, strategy-to-contract module.
	***InContextOf*** In addition to the permitted connections defined in the core GSN definition (Section 1.2), in the modular GSN extension the following additional connections are permitted: goal-to-away goal, goal-to-away context, goal-to-module, strategy-to-away goal, strategy-to away context and strategy-to-module.

Figure 24: Extensions to Core GSN Relationships to support Modularity

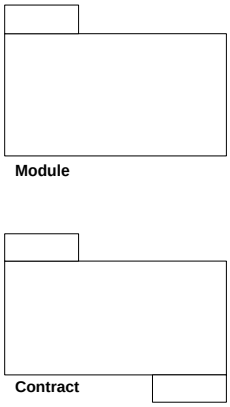
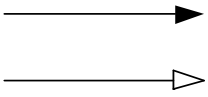
	***Module*** and ***Contract Module*** symbols are used in Module view to represent the referenced module of argument without displaying the content of the argument. The arguments represented by these symbols are not necessarily documented using GSN.
	***SupportedBy*** and ***InContextOf***, when used in the module view can represent one or more support/context relationship(s) between the elements within the modules.

Figure 25: Extensions to the Core GSN Notation to support Module View

B1.3 Notation Interpretation

B1.3.1 Intra-Module Notation

B1.3.1.1 The core GSN elements defined in Sections 1.2 and B1.2 above are intended to be combined to represent logical structures. The notation interpretation for core entities 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 module. *Away goals* cannot be (hierarchically) decomposed and further supported by sub-entities within the current module; rather, decomposition needs to occur within the referenced module.

B1.3.1.2 Arguments supported by another module can be indicated in a number of ways. Figure 26 illustrates a firm relationship by which the parent *goal* is supported by a specific *goal* in the referenced *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*.

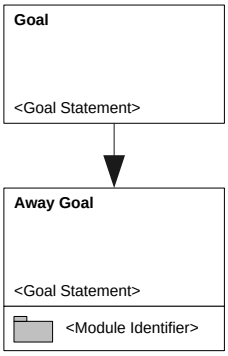


Figure 26: Use of 'Away Goals'

B1.3.1.3 Figure 27 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*.

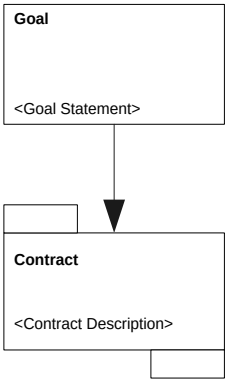


Figure 27: Use of Contract

B1.3.1.4 An alternative approach is illustrated in Figure 28. The *contract module* instantiating the support relationship is not specified. Here, the relevant higher-level argument abstraction (e.g. module view) should be referred to, which will indicate where the required *contract* details are specified.

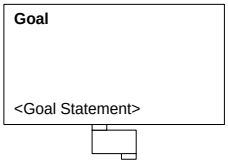


Figure28: Use of Unspecified Contract

B1.3.1.5 Where a *module reference* element is shown in support of a parent *goal* as illustrated in Figure 29 below, this signifies that the parent *goal* is supported by the entire argument made in the referenced *module*.

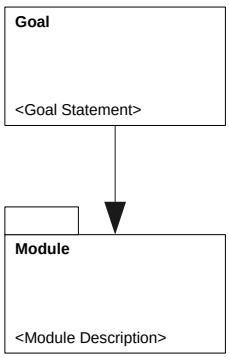


Figure 29: Use of a Module

B1.3.1.6 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 1.3 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 *module* as context for the *goal* or *strategy* he is currently working with. Use of *away goals* to replace justification for GSN *goals* and *strategies* is illustrated in Figure 30:

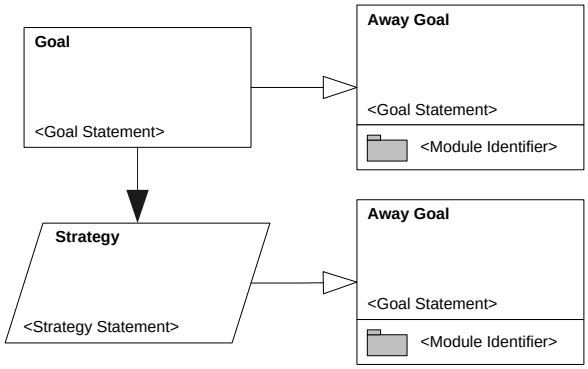


Figure 30: Use of 'Away Goals' to replace Justification

B1.3.2 Inter-Module Notation

B1.3.2.1 It is useful to represent the abstracted structure of an argument in a module view. The process of abstraction hides the detailed structure of the argument. *Goals*, *strategies, solutions* and *context* are not shown in the module 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.

B1.3.2.2 Figure 31 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 as in module 1. Similarly, 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 modules.

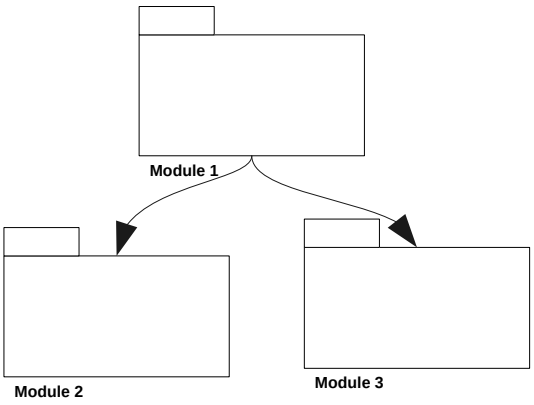


Figure 31: 'Supported By' Relationship between Modules

B1.3.2.3 *Contract modules* can be used in the support relationship between *modules* to aid decoupling as shown in Figure 32. This de-coupling 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.

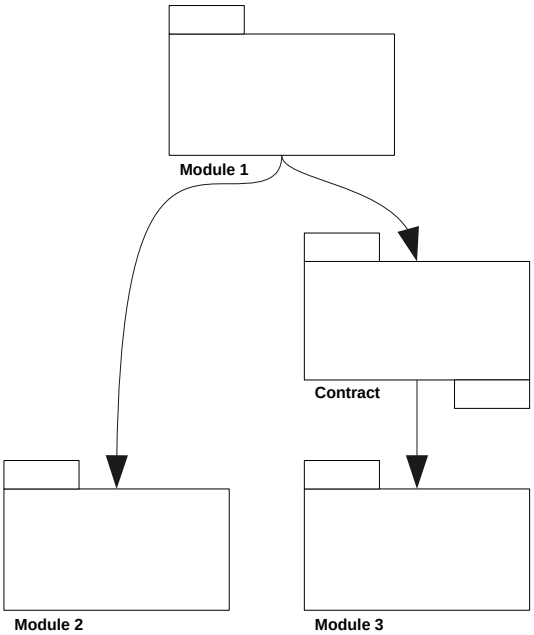


Figure 32: Use of Contract

B1.3.2.4 The *InContextOf* relationship between the two *modules* in Figure 33 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 2:

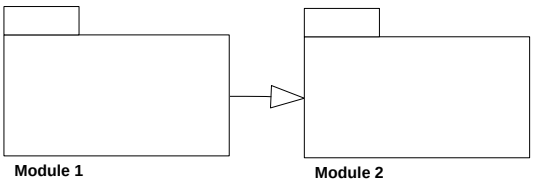


Figure 33: 'In Context of' Relationship between Modules

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 he has 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 several ways: top-down, bottom-up or any combination of the two.

2.1.3This 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.

2.2 Guidance on the Layout of Goal Structures

2.2.1 In this section, we present 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 is clear from Section 0.3 above (and, indeed, from the language used throughout this Standard to describe relationships between claims), the claims made in GSN *goal* elements 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*.

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.

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 child *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 GSN diagrams 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.

2.2.11 The {Element Identifier} in a GSN element (whether in the core notation or the modular extensions) is optional.
2.2.12 The {Element Statement} is mandatory, and should be expressed according to the advice given in Section 1.4 above.

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 [**5**]. 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 [**5**] 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 34 illustrates this six-step process, which is recursive. Having first identified a claim and represented it using a GSN *goal* (step 1), we make an explicit statement of the context in which it is valid (step 2). We then identify a *strategy* to support it (step 3) and justify this *strategy* (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).

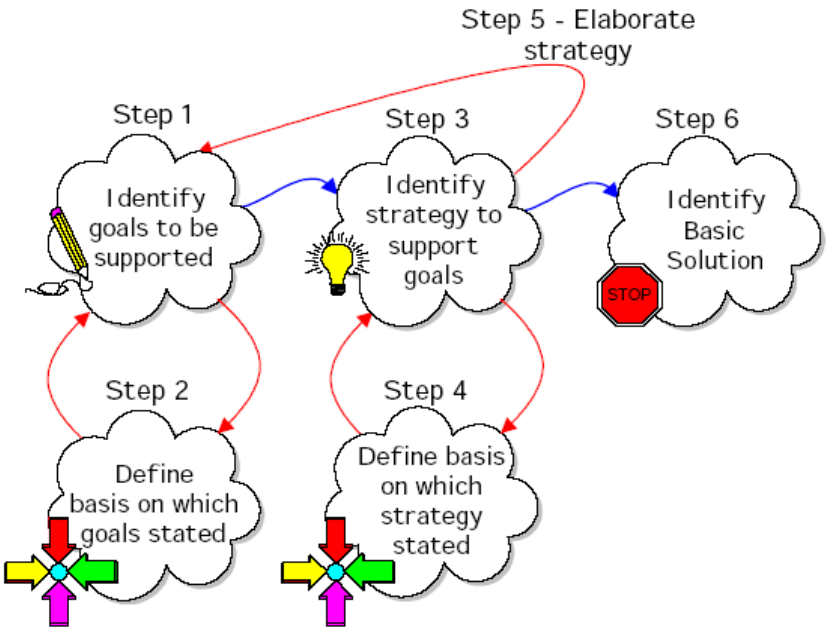


Figure 34: Six-Step Process for Developing Goal Structures

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's drawing his conclusions at too low a level and precludes the demonstration of the derivation of the top-level claim from that fundamental objective. Figure 35 introduces the top *goal* of the running example used to illustrate the top-down development of a goal structure in this section:

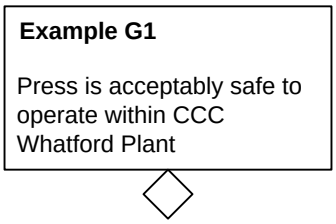


Figure 35: Top Goal of Running Example

2.3.3 Step 2: Definition of the Basis on which Goals are Stated

2.3.3.1 A claim made in a goal structure (or, indeed, in any other argumentation 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 he is 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 (see Section 1.3 above). Figure 36 illustrates the association of *contex*t with a *goal*, to clarify concepts introduced in the claim:

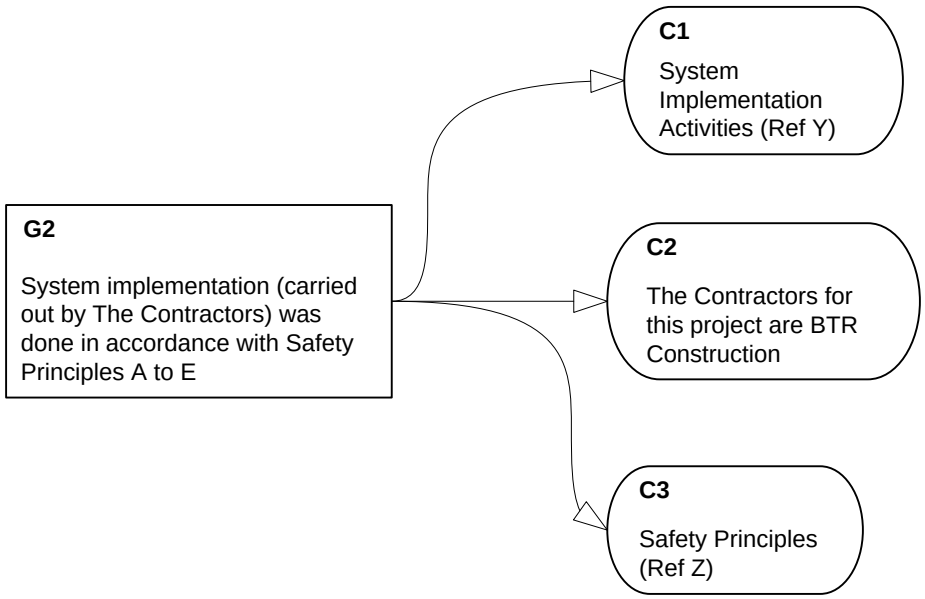


Figure 36: Association of Additional Contextual Information

2.3.3.3 In Figure 36, 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, as discussed in Section 1.3 above, contextual information associated with a claim made in a particular *goal* is understood to be in scope for all sub-goals of 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 37 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:

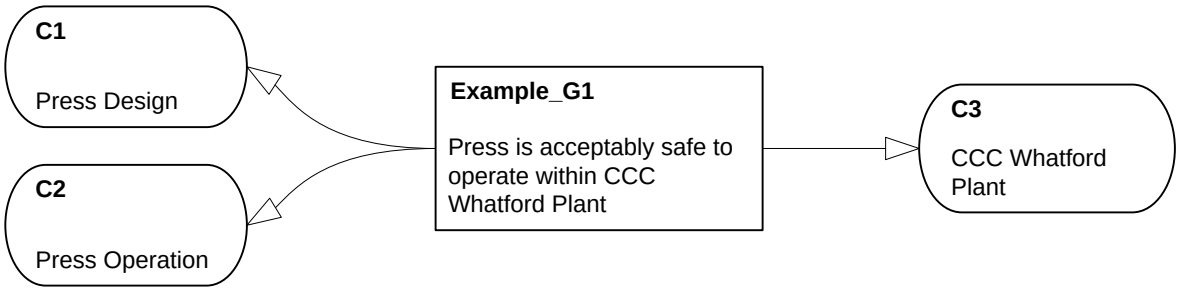


Figure 37: Example with Contextual Explanation

2.3.3.7 In Figure 37, 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. We have left the concept "acceptably safe" to be defined through the supporting argument.

2.3.4 Step 3: Identification of 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 38 illustrates this approach:

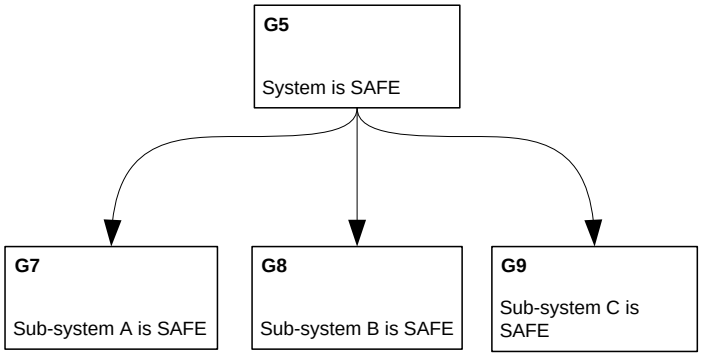


Figure 38: Divide and Conquer Goal Decomposition

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 39 illustrates this approach:

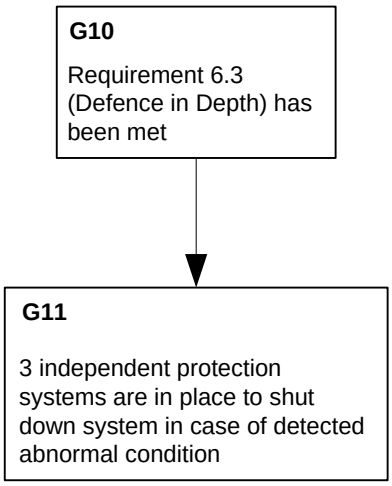


Figure 39: Interpretation, or Particularisation, of a Goal

2.3.4.4 As outlined in Section 1.3 above, 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 sub-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 40 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 argumentation which are being put forward to support the claim made in Goal G1:

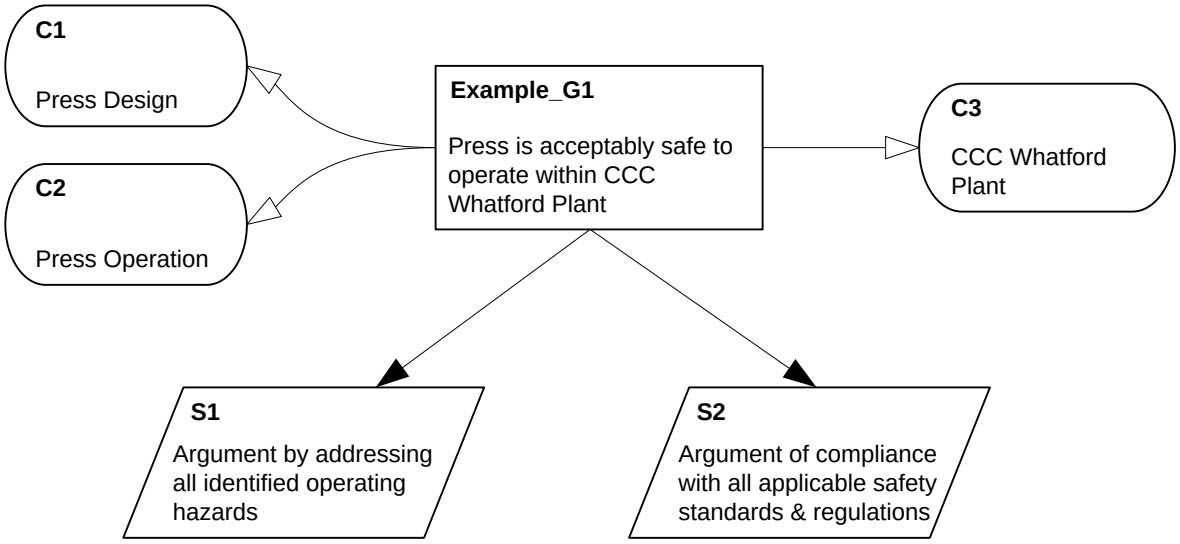


Figure 40: Example with Top-Level Strategies

2.3.5 Step 4: Definition of 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 40 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 facts about the system, its operating context, users or environment that the *strategy* depends on (see Section 1.3 above). *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. Section 1.3 describes the representation of *assumptions* and *justifications* in GSN.

Example

2.3.5.3 Continuing the development of the goal structure, Figure 41 shows the contextual information necessary to clarify Strategies S1 and S2:

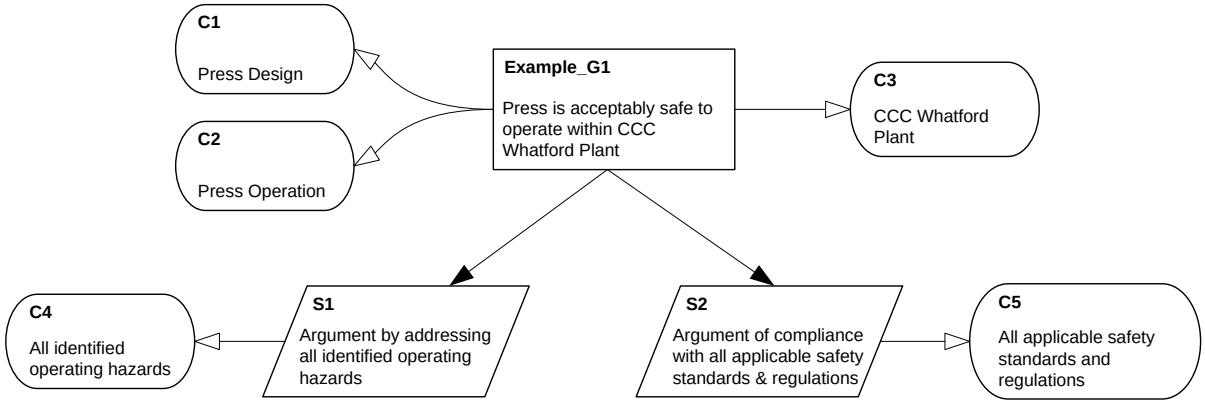


Figure 41: Example with Contextual Evidence to Clarify Strategies

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, he should attach appropriate *justifications* 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 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 sub-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 sub-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 42 shows the elaboration of the *strategies* defined in Figures 40 and 41. 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).

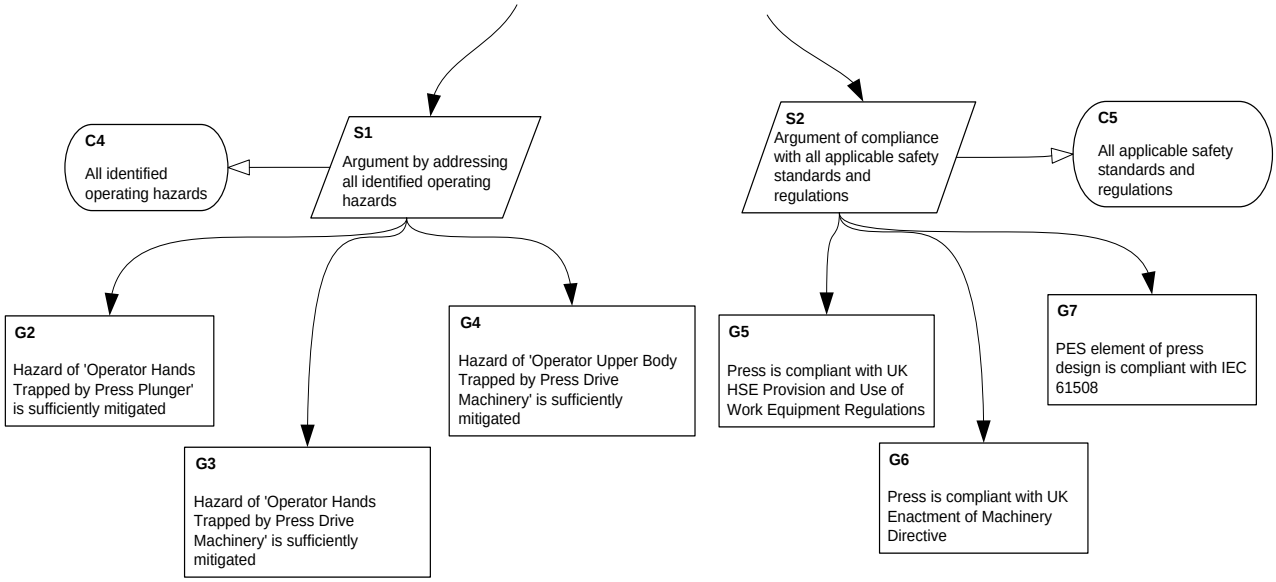


Figure 42: Elaboration of Strategies

2.3.6.6 The goal structure continues to be developed in this way until it is clear that no further decomposition into sub-goals is necessary and the *goal* can be directly supported by appeal to some evidence artefact (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 (see Section 1.3 above). The *solution* provides a reference to some evidence artefact. Figure 43 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 42). 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.

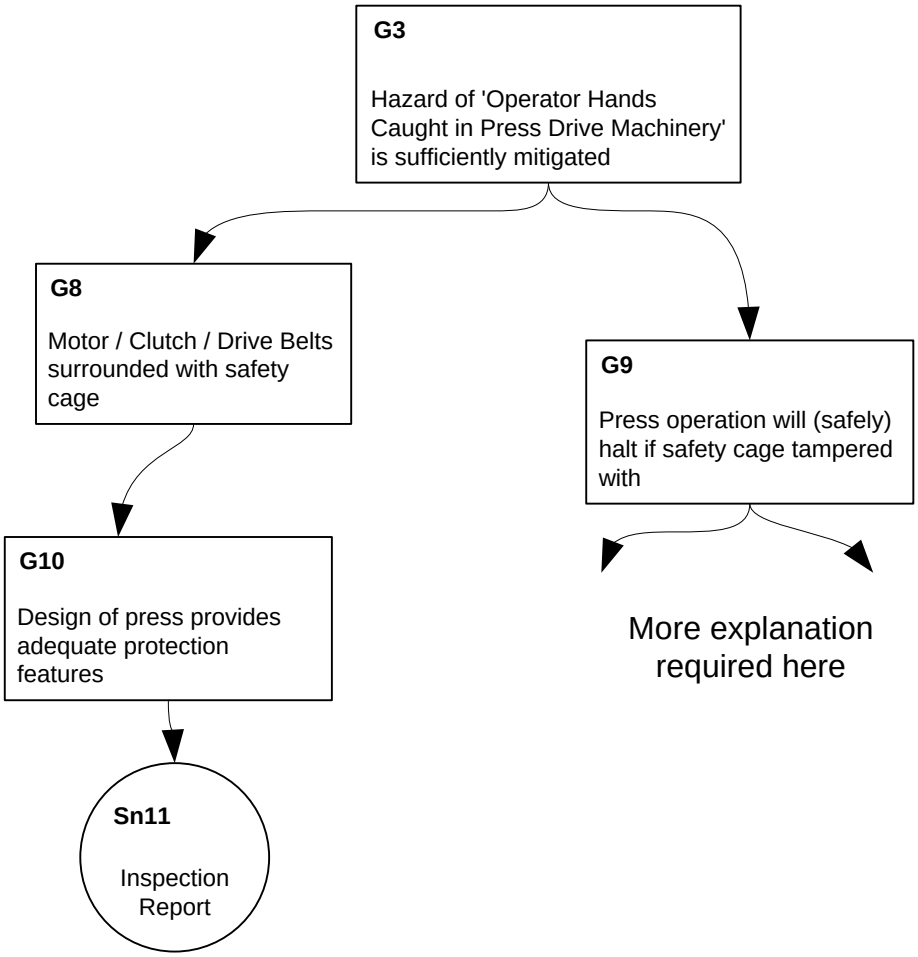


Figure 43: Reference to Evidential Support

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 artefact 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 we can't close out the argument?

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 he requires 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 (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 sub-goals that are supported by the evidence assertions;
4. Describe how each layer of sub-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 sub-goals.

Figure 44 shows these steps graphically:

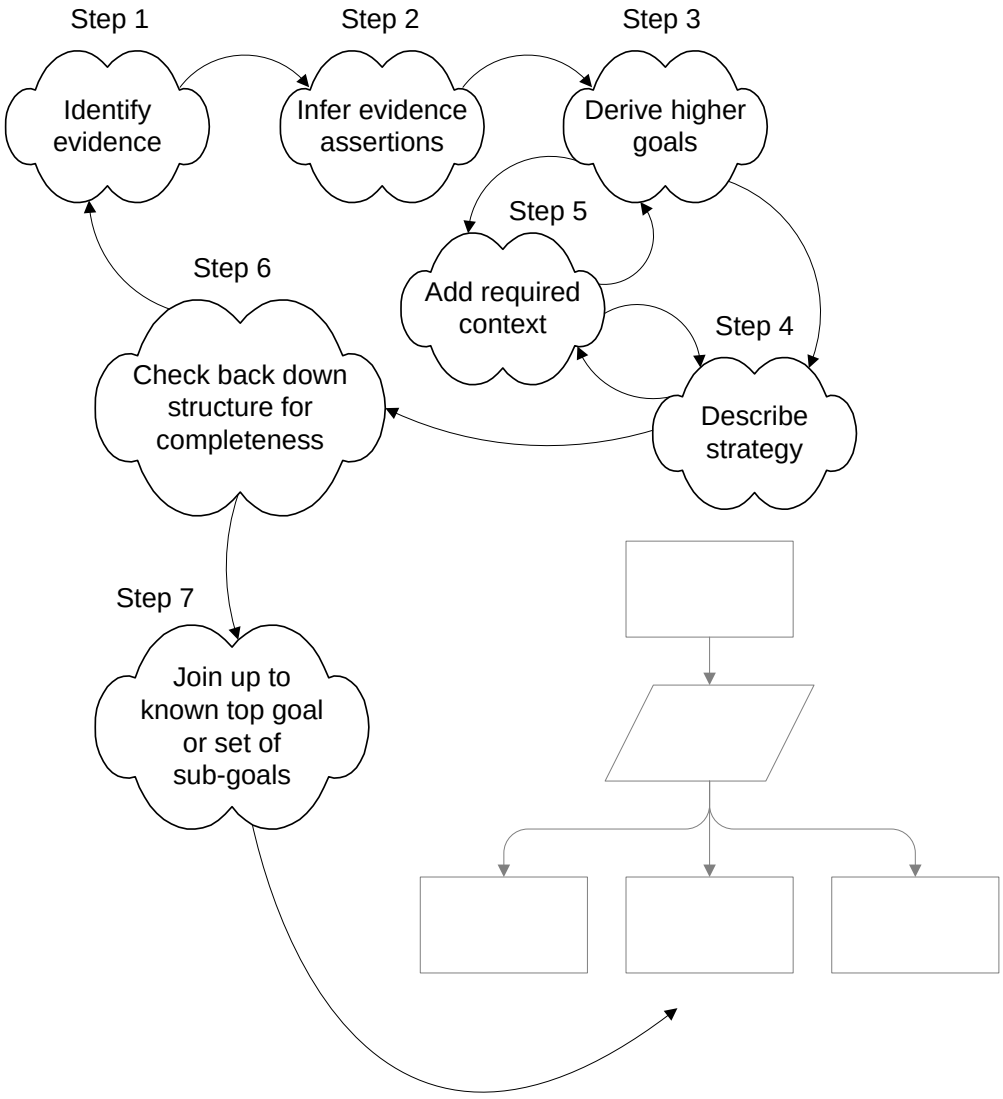


Figure 44: Bottom-Up Process for Developing Goal Structures

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 relies 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 45:

Figure 45: Typical Solutions Derived from Evidence

2.4.2.2 Having created evidence artefacts 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 artefact will suggest a claim, which is represented as a bottom-level 'evidence assertion' *goal* in the assurance argument (see Section 2.3.7), 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 46 demonstrates the inference of evidence assertion sub-goals directly from solutions in GSN:

Figure 46: Evidence Assertion from Sub-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 (Section 2.4.4) 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 47 illustrates this approach:

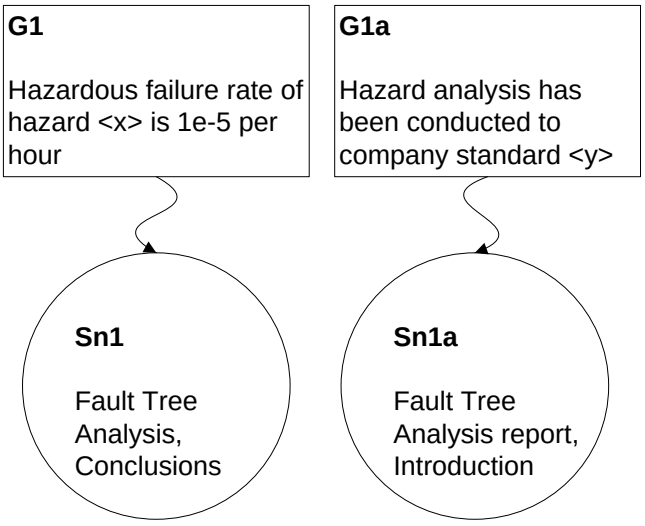


Figure 47: Multiple Evidence Assertion Sub-Goals Inferred from Similar Solutions

2.4.4 Bottom-Up Step 3: Adding Higher Sub-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 48 illustrates adding a higher-level sub-*goal*:

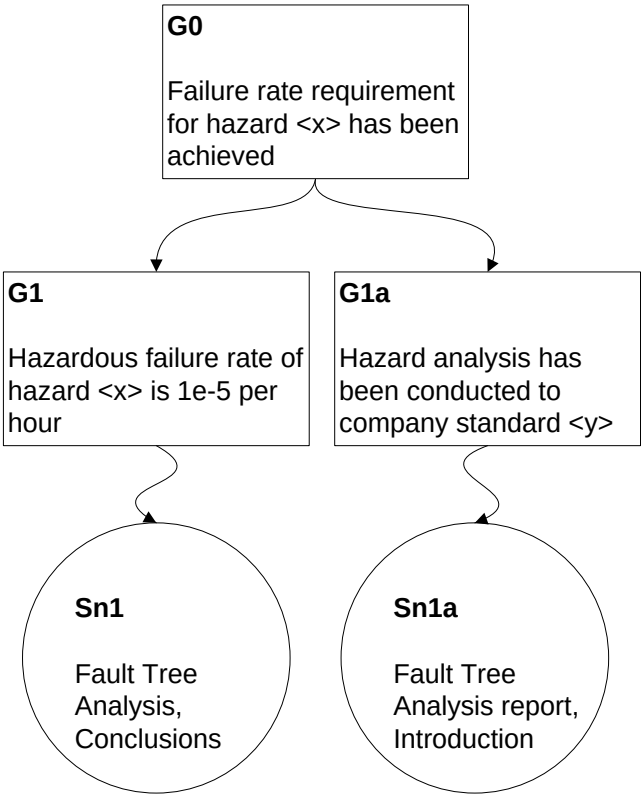


Figure 48: Adding a Higher-Level Sub-Goal

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 argumentation 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 48 – 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: Describe Strategy for Goal-Decomposition

2.4.5.1 When deriving a higher-level *goal* that is supported by its sub-goals, it can be helpful to describe how those sub-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 sub-goal decomposition is achieved. Figure 49 shows the addition of a strategy to describe the step between parent *goal* and sub-goals:

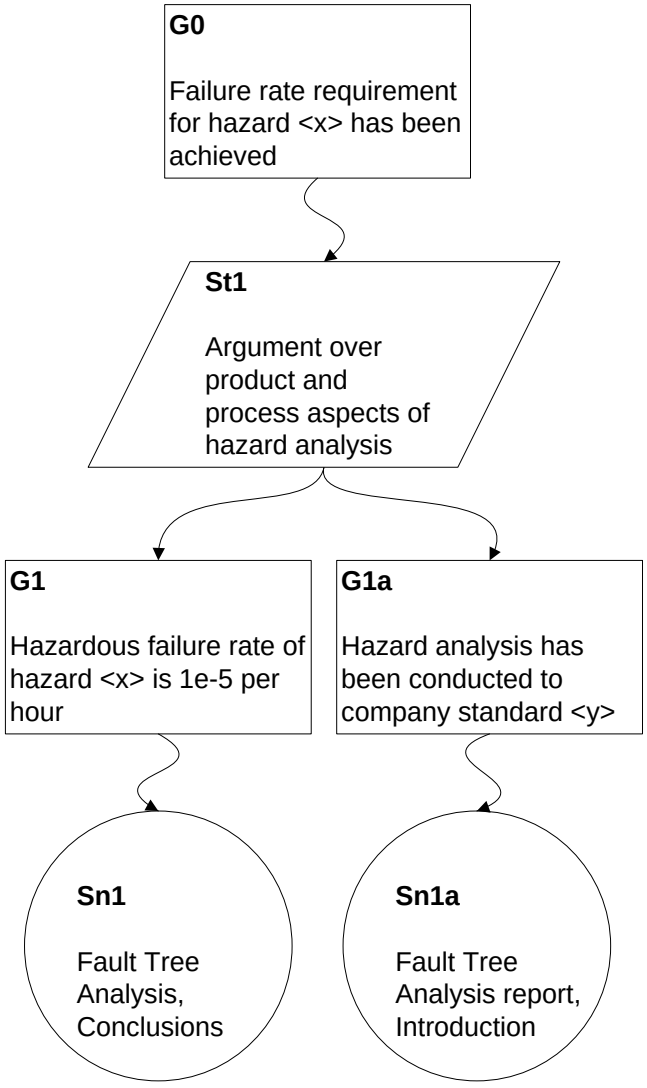


Figure 49: Describing the Strategy for Goal Decomposition

2.4.5.2 Should the decomposition strategy be 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 50 shows the addition of *contexts* to the parent *goal*:

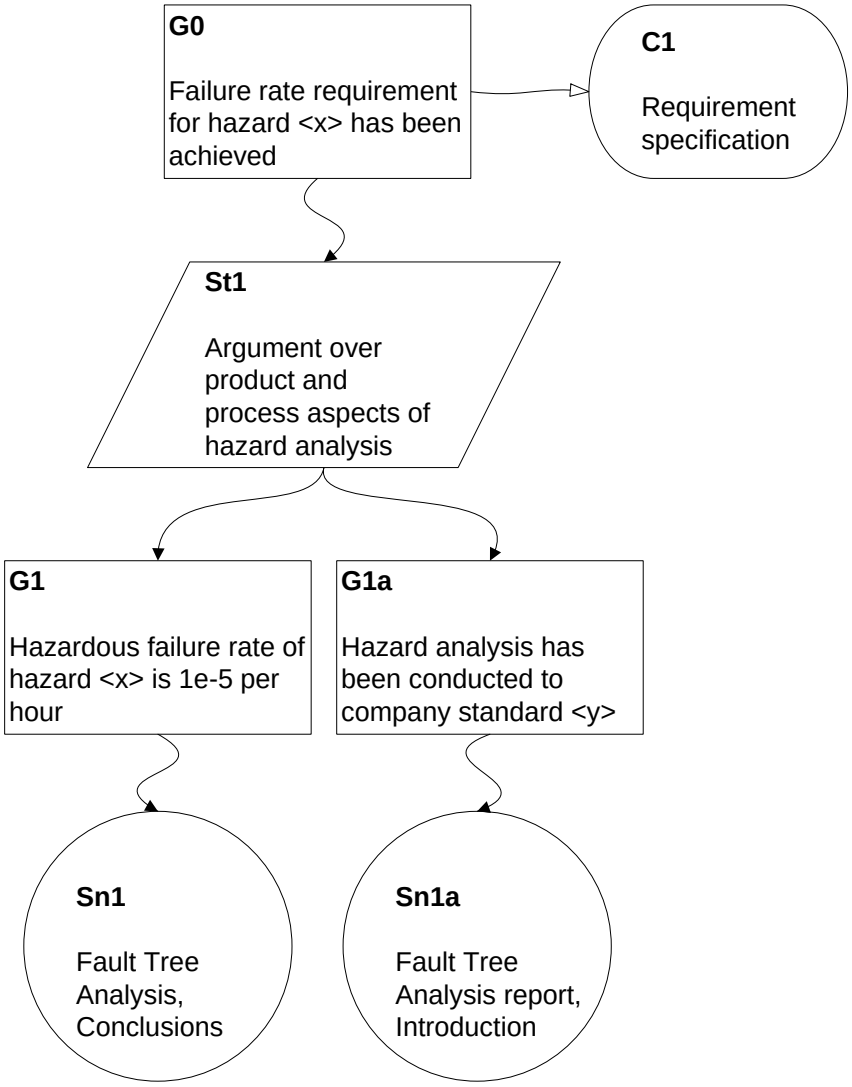


Figure 50: Adding Contextual Information

2.4.6.2 For example, an appeal to FTA evidence this 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: Check Back Down the Goal Structure

2.4.7.1 Each time a parent *goal* is created, the author should re-examine the supporting sub-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 sub-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 Figures 46-50, 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: Incorporate Bottom-Up Goal Structure into 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 51 illustrates this connection:

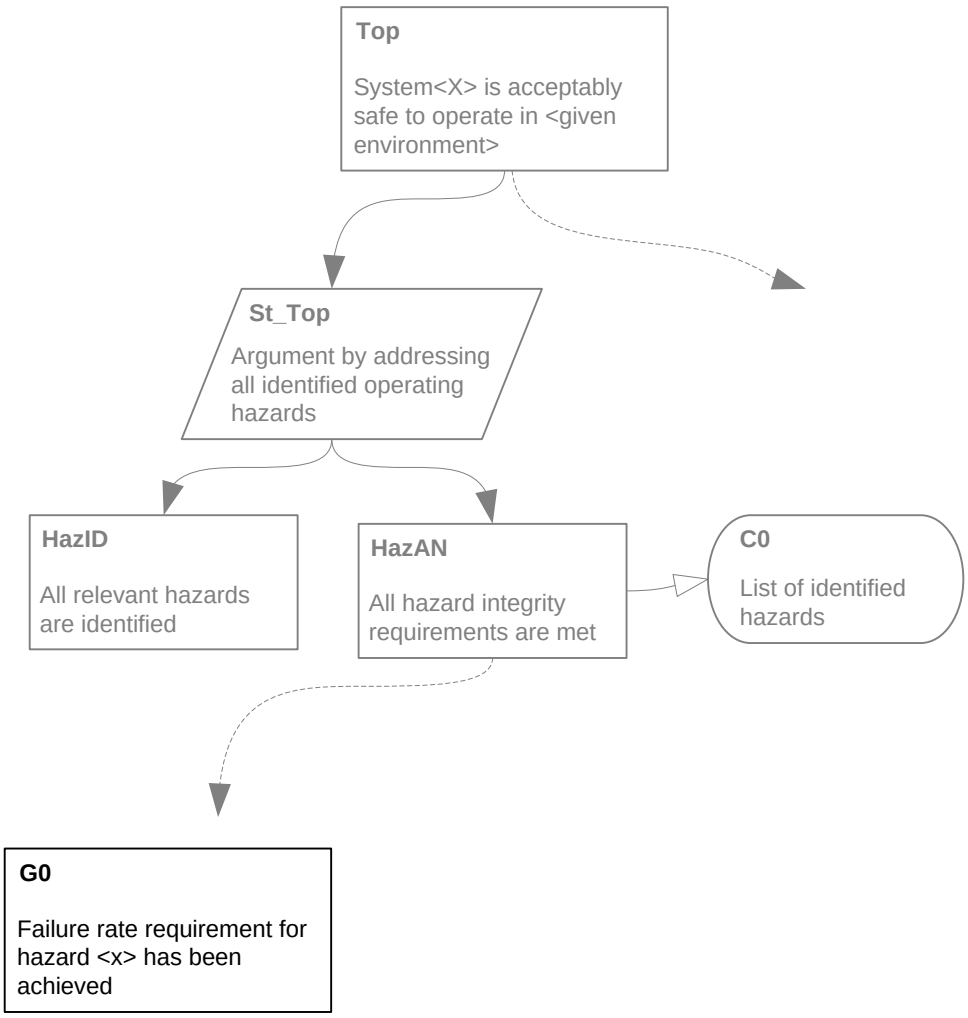


Figure 51: Joining the Bottom-Up Goal Structure to a Higher Fragment

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 rely on contextual 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 he has, and scope his claims accordingly. Where possible, claims which are potentially undermined by shortcomings in one evidence artefact should appeal to more than one evidence artefact for support.

2.5 Avoidance of Common Errors in Creating Goal Structures: Part 1 – Language Issues

2.5.1 Introductory

2.5.1.1 The guidance presented in Sections 2.5 and 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 argumentation, while others arise from the use of argumentation 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 we have identified the most commonly encountered issues, Sections 2.5 and 2.6 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 his 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 a *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 *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 artefacts – 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 of argumentation:

- **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.
- **Allow 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 (see Section 1.2 above). *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 artefact provides a direct solution.

- **Allow contexts to act as references**. As defined in Section 1.3 above, *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" [**1**]. 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 dialectal 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.

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 Avoidance of 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 his 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 52 illustrates this incorrect use of a GSN *context* element to support a claim made in a goal:

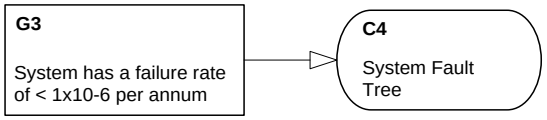


Figure 52: Incorrect Use of Context (as a Solution)

2.6.2.2 Here, Context C4 is incorrectly associated with Goal G3 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 G3 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 53, for example, the statement "System X has no common-mode failures" would be more appropriately rendered as an *assumption* than as a *context*:

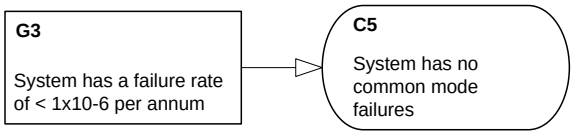


Figure 53: Incorrect Use of Context (as an Assumption)

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 his working' between steps in a proof, thus making it difficult to see how he 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 54, for example, it is difficult for the reader to see the relationship between G1 and G2, since the reasoning by which inclusion of a safety cage justifies a claim of acceptable system safety is not clear:

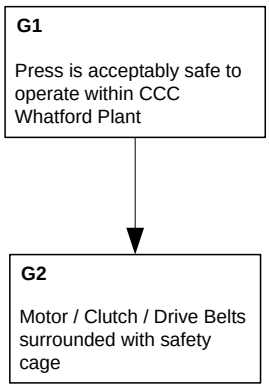


Figure 54: An Inductive Leap

2.6.4.3 In order to ensure that the reader can follow the logical thread of the argument he is making, the author should add some additional goals between G1 and G2, to serve as 'stepping stones' the reader can follow:

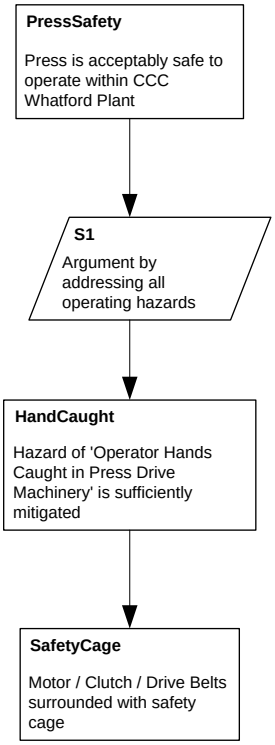


Figure 55: Intermediate Goal as a 'Stepping Stone'

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 56:

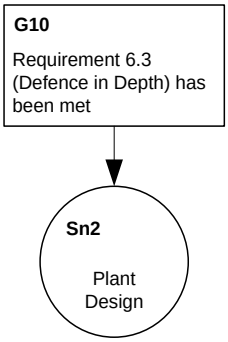


Figure 56: 'Jumping' to a Solution

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 will be confused 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 sub-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. 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 The most obvious place for review in the system lifecycle is 'pre-operational', i.e. just prior to the system's being approved for entry into service. For example, early review of the strategy adopted in an assurance argument could be very useful. However, 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.

2.7.2.2 The most compelling staged reviews will involve representatives from the acceptance authority and any other key stakeholders. It is often not possible to get an acceptance authority to confirm 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 the acceptance authority, 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 57 illustrates a staged approach to the review of assurance case arguments, derived from [**6**]:

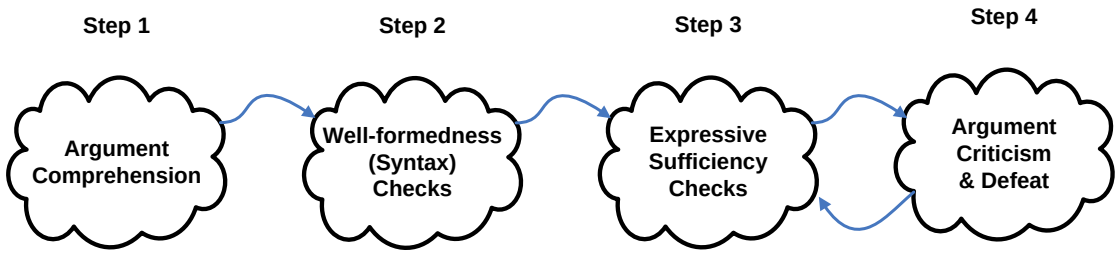


Figure 57: Staged Argument Review Process

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., we cannot check the well-formedness of an argument before we fully comprehend its structure) 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 stage 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.

2.7.4.1.2 Where the assurance case has been presented textually, it can be useful to mark the text up with coloured highlighters identifying each element in the argument (evidence, assumptions, claims etc.). Having identified the essential elements of the assurance case, it is then necessary for the reviewer to 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 annotate the document further with cross-references. At this point, it can often be useful to attempt to re-represent the argument using GSN. Constructing such a representation of the argument structure can be the 'acid test' of whether the reviewer really understands the nature of the argument being presented.

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.
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 sub-goals related to 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 (See Section 1.3 above). 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

2.8.4.4.1 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.
- **Relevance** – how relevant is a particular piece of evidence or line of argumentation 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 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 all of 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 thoroughly reviewed by suitably competent and experienced personnel? This principle is already enshrined in several safety standards; for example, RTCA/DO 178B requires independent review of software items developed to high Design Assurance Levels (DALs) [**7**].
- 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-178B 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. This has been recognised by the Defence Standard 00-56 (Issue 4, Part 2 Paragraph 9.5.6) [**8**]:

> *Throughout the life of the system, the evidence and arguments in the Safety Case should be challenged in an attempt to refute them. Evidence that is discovered with the potential to undermine a previously accepted argument is referred to as counter-evidence. The process of searching for potential counter-evidence as well as the processes of recording, analysing and acting upon counter-evidence are an important part of a robust Safety Management System and should be documented in the Safety Case.*

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.

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.

ANNEXES TO PART 2

A2 GUIDANCE ON PATTERN EXTENSIONS

A2.1.1 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.

A2.1.2 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);
- **Entity Abstraction**, which allows a distinction to be made between classes and instances of GSN element-types.

A2.1.3 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.

A2.1.4 Entity 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".

A2.1.5 Section A1 above defines the GSN symbology introduced to support structural abstraction: the optionality and multiplicity annotations attached to the *SupportedBy* relationship, and the *option* symbol, which is used to represent choices between lines of argumentation used to support a particular *goal*.

A2.1.6 Multiplicity relations can be combined with optionality relations. Placing multiplicity annotations on the 'supported by' symbols prior to the GSN *option* symbol describes a multiplicity over all of the optional relations. Placing a multiplicity symbol on individual optional relations (i.e. just prior to the sink) describes a multiplicity over that relation only.

A2.1.7 It is useful to provide an annotation next to the option symbol denoting the nature of the choice to be made – e.g. '1 out of n' or '2 out of 3'.

B2 GUIDANCE ON MODULAR EXTENSIONS

Text to be supplied.

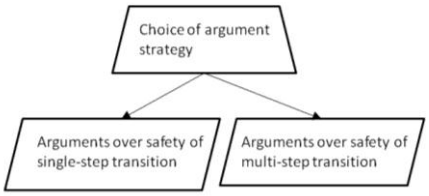
C2 OTHER EXTENSIONS TO GSN

C2.1 Introductory

C2.1.1 From time to time, elements other than those defined in Section 1 of this Standard may be encountered in GSN diagrams. These have formed part of the notation as it has evolved, and are currently supported by at least one off-the-shelf GSN editing tool. Figure 58 illustrates the elements used, and explains the concepts they are intended to represent.

C2.1.2 A number of these elements derive from the use of GSN in requirements capture and analysis.

C2.1.3 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.

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 *option* 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*

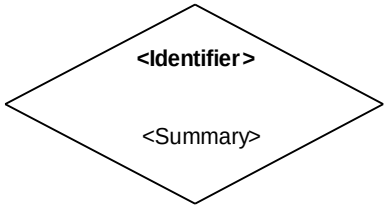
Stakeholder 	This is a form of *context* symbol which is used to indicate one of the stakeholders associated in some way with the *goal* to which it is attached. Example: *Installation Contractor (ABC Cabling Ltd)*
Problem 	This is a form of *context* symbol which is used to indicate that there is a 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. 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.

Figure 58: Non-Modular Extensions to GSN

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.

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.

REFERENCES

[1] *Shorter Oxford English Dictionary* 6th edn (2007)

[2] 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.

[3] Toulmin, S.: *The Uses of Argument* (1958; 2nd edn, 2003)

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

[5] Kelly, T.: 'Arguing Safety: A Systematic Approach to Managing Safety Cases', D.Phil Thesis, University of York (1998). Available for download from http://www-users.cs.york.ac.uk/~tpk

[6] 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)

[7] RTCA/DO-178B *Software Considerations in Airborne Systems and Equipment Certification* (1992)

[8] UK Ministry of Defence, *Interim Defence Standard DS 00-56 (Issue 4), Safety Management Requirements for Defence Systems* (June 2007)

