DRAFT GSN STANDARD

VERSION 1.0

Draft released to the GSN User Community and other Interested Parties for Comment

Consultancy Period: May 19th to August 27th 2010

Your comments on this interim, incomplete Draft Standard would be warmly welcomed by the Drafting Committee. We have set up a Google Wave to handle comments and discussion about the Standard. To join the wave, please email details of your Google Wave account or your email address to katrina.attwood@cs.york.ac.uk. If you are unable to access the Wave, please email comments to the above address, and they will be added to the Wave for you.

The Drafting Committee will meet at the end of the Consultancy Period, and revise the draft in the light of comments received. Our intention is to publish Issue 1 of the Standard by 1st November 2010.

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 the current best practice in 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 2010. The document history on page ii outlines the recent history of the collaboration, and a list of contributors to the Standard is provided on page iii.

DOCUMENT HISTORY

Version	Issued to	Date	Purpose
0.3	Review Committee	2nd May 2010	For review prior to general circulation
1.0	User Community and interested parties	19th May 2010	For consultation

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 Peacock

Clive Pygott

Mick Warren

Phil Williams

CONTRIBUTING ORGANISATIONS

AACE Ltd

Adelard LLP

Altran Praxis Ltd

BAE Systems Ltd

Columbus Computing Ltd

CSE International Ltd

ERA Technology Ltd

General Dynamics UK Ltd

LR Rail Ltd

Origin Consulting (York) Ltd

RPS Group Ltd

SafeEng Ltd

Selex-Galileo Ltd

Thales Ltd

0.1.1 UK Ministry of Defence

0.1.2 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. The Goal Structuring Notation is a graphical argumentation notation that can be used to explicitly document the individual elements of any argument (requirements, claims, evidence and context) and (perhaps more significantly) the relationships that exist between these elements (i.e. how individual requirements are supported by specific claims, how claims are supported by evidence and the assumed context that is defined for the argument). Arguments documented using GSN can help provide assurance of critical properties of systems, services and organisations (such as safety or security properties).

The standard is grouped in four parts, as follows:

- **Part 0: Introduction and Concepts.** 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.** 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, 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.** This part provides informative guidance on the effective use of GSN to create and evaluate structured arguments.
- **Part 3: Web-Based Resources.** This part provides additional guidance on the use of GSN, including further examples of goal structures and catalogues of existing argument patterns.

1 INTRODUCTION AND CONCEPTS

1.1 Introductory

This section provides information about GSN sufficient to enable a novice user to read and understand a goal structure represented using the notation without recourse to the other parts of the standard.

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 (such as a safety case). The role of arguments in assurance cases is explained in the next section.

1.2 Use of Arguments in Assurance Cases

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

A safety case can be defined as:

> *the argument and supporting evidence that a system, service or organisation is acceptably safe for a given application in a given environment.*

In order that safety cases can be presented, reviewed, discussed and accepted amongst stakeholders it is necessary that they are clearly documented. The documented argument of the safety case should be structured to be comprehensible to all safety case stakeholders. It should also be clear how the evidence is being claimed to support this argument. By appealing to core concepts of argumentation (as explained in the next section), GSN helps address these objectives.

1.3 What is an Argument?

In the assurance domain an '**argument**' is defined as "a connected series of statements or reasons intended to establish a position…; a process of reasoning" [1]. In attempting to persuade others of a position, we cite reasons why a claim should be accepted as **true**. These reasons are described as the **premises** of the argument, and the claim they support as its **conclusion**. These terms can be used to define the 'normal form' of an argument as:

Premise

Premise

Premise

So, Conclusion

This form reduces argument to its most primitive building blocks, for example:

Premise: All complex systems are susceptible to failure.

Premise: Failures can lead to accidents.

Therefore,

Conclusion: Accidents can occur in complex safety-critical systems.

The terms 'premise' and 'conclusion' are relative. The premise of one reasoning step (e.g. that "All complex systems are susceptible to failure") may itself need further reasoning support and will become the conclusion of a subsequent supporting argument. This gives rise to hierarchical argument structures ('chains of reasoning') in which arguments are established by the composition of a number of (premise-conclusion) reasoning steps in order to support an overall conclusion.

At the heart of GSN is the explicit documentation of these hierarchical argument structures. In the next section, the key elements of the notation are explained.

1.4 The Goal Structuring Notation (GSN)

GSN is a graphical argumentation notation which can be used to explicitly document the elements of any argument (claims, evidence and context) and - perhaps more significantly - the relationships that exist between them (i.e. how claims (conclusions) are supported by other claims (premises), how claims are supported by evidence, and the assumed context which is defined for the argument). 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].

The purpose of GSN is to document how *goals* (the *conclusions* of the argument) are said to be supported by *sub-goals* (the *premises* of the argument). It can then be shown how these *sub-goal*s are being said to be supported by subsequent supporting goals. Figure 1 shows an example goal in GSN.

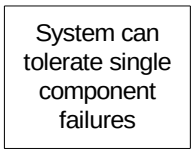


Figure 1: An Example Goal

Where evidence is said to exist to support the truth of the claimed goal 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

When documenting how *goals* are said to be supported by *sub-goals* it can be useful to document the *reasoning step* – i.e. the nature of the argument that connects the goal to its sub-goals. This is done in GSN by documenting the connecting argument *strategy.* Figure 3 shows an example strategy in GSN.

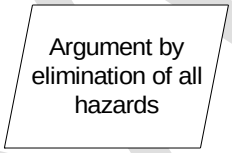


Figure 3: An ExampleStrategy

When documenting a *goal* it can also be important to capture the *context* in which that claim should be interpreted. This is done in GSN by documenting context. Figure 4 shows an example context in GSN.

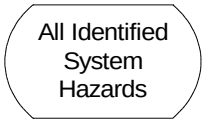


Figure 4: An Example Context

Goals, strategies, solutions and context form the principal elements of GSN. (Other element types exist and are explained in Part 1.)

When the elements of GSN are connected together they are said to form a 'goal structure'. Figure 5 shows an example goal structure. Goal structures document the chain of reasoning in the argument (through the visible decomposition of claimed *goals* and description of argument *strategies*), how this argument is supported by

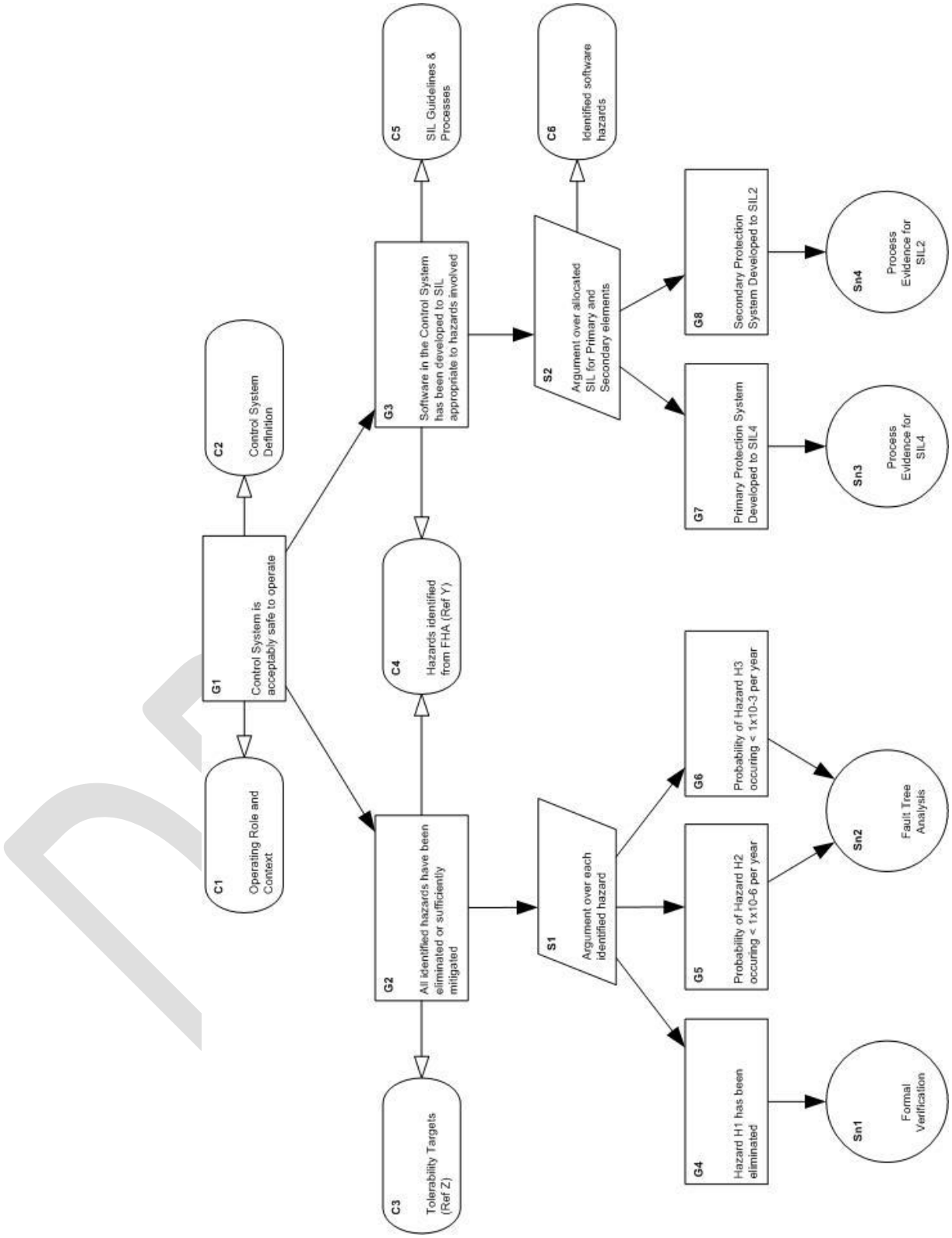


Figure 5: An Example Goal Structure

evidence (through *solutions*) and clearly capture the *context* in which the claimed goals of the argument are being put forward.

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 project stakeholders (e.g. system developers, engineers, independent assessors and certification authorities). In turn, this improves the quality of the debate and discussion amongst the stakeholders and can reduce the time taken to reach agreement on the argument approaches being adopted. For example, using the goal structure provided in Figure 5, it would be reasonable to challenge 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.

2 DEFINITION OF GOAL STRUCTURING NOTATION

2.1 Introductory

This section provides a normative definition of the Goal Structuring Notation. The core elements of the notation are introduced in Section 1.2. Section 1.3 describes the interpretation of permitted combinations of these elements. Given that Goal Structuring Notation comprises both the element nodes and the language they contain, Section 1.4 defines the rules which apply to the language structures used to reflect the logical structures established by the symbology. Extensions to the core GSN to support the development of generic argument patterns and modularised arguments are defined in Annexes A1 and B1.

2.2 Notation

The following core symbols are used in GSN:

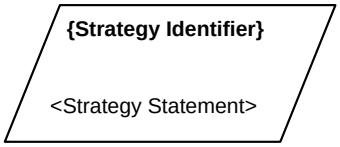
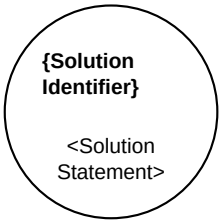
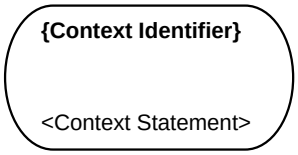
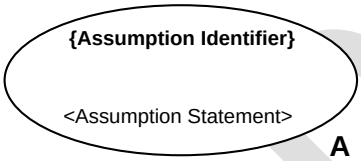
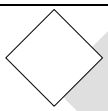
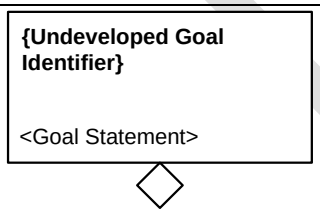
- Goals
- Strategies
- Solutions
- Contexts
- Assumptions
- Justifications

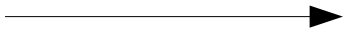
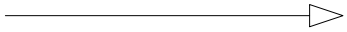
These are linked using the following types of relationships:

- Supported by
- In context of

The rest of this section provides the definition and rendering of these symbols and relationships. The meanings of structures combining these elements are further explained in Section 1.3. 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.

{Goal Identifier} <Goal Statement>	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 one or more goals and another goal.
	A **solution**, rendered as a circle, presents a reference to evidence 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 latter 'A' at the bottom-right, presents an intentionally unsubstantiated statement.
	**Undeveloped entity**, rendered as a hollow diamond, indicates that a line of argument has not been developed. It can apply to goals (as below) and strategies.
	2.2.1.1.1.1.1.1 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.

	**Supported by**, rendered as a line with a solid arrowhead, declares an inferential[3] or evidential[4] relationship. Permitted connections are: goal-to-goal, goal-to-strategy, goal-to-solution, strategy-to-goal.
	**In context of**, 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

2.3 Notation Interpretation

The core GSN symbols defined in Section 1.2 above are intended to be combined to represent logical structures, known as 'goal structures'. The previous section defined permitted relationships between the elements of GSN.

Figure 6 shows the most basic relationship represented in goal structures; inferences between goals:

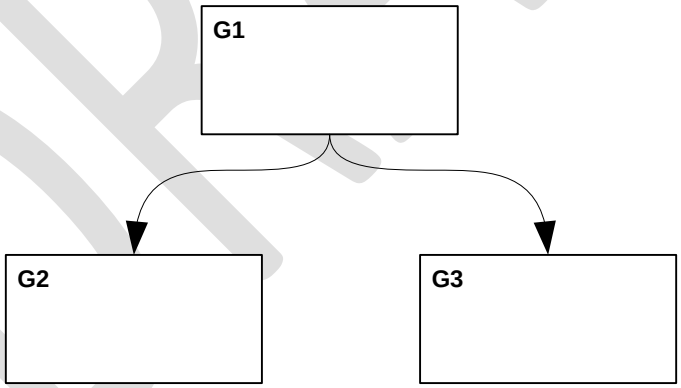


Figure 6: Supporting Goals with Sub-Goals

This specific structure asserts that if the goals G2 and G3 are true, this is sufficient to establish that 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.

[3] Inferential relationship: presenting a declared inference between the goals of the argument.
[4] Evidential relationship: presenting a declared relationship between a goal and an evidence item by which the goal is substantiated.

The structure shown in Figure 7 also asserts that if goals G2 and G3 are true, this is sufficient to establish that 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.

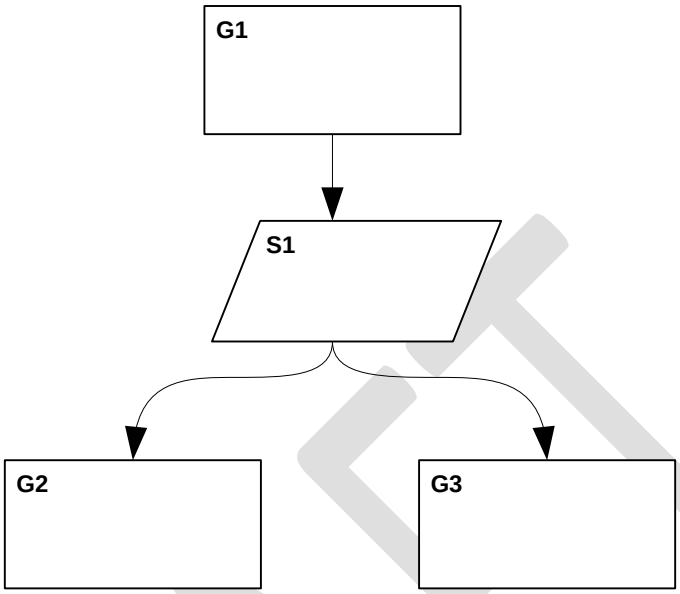


Figure 7: Adding Strategy

In some cases, more than one argument approach may be adopted in support of a parent goal. Figure 8 represents a relationship of this type, by which the separate contributions of the goal groupings (G2, G3) and (G4, G5) are made explicit in Strategies S1 and S2 respectively. 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.

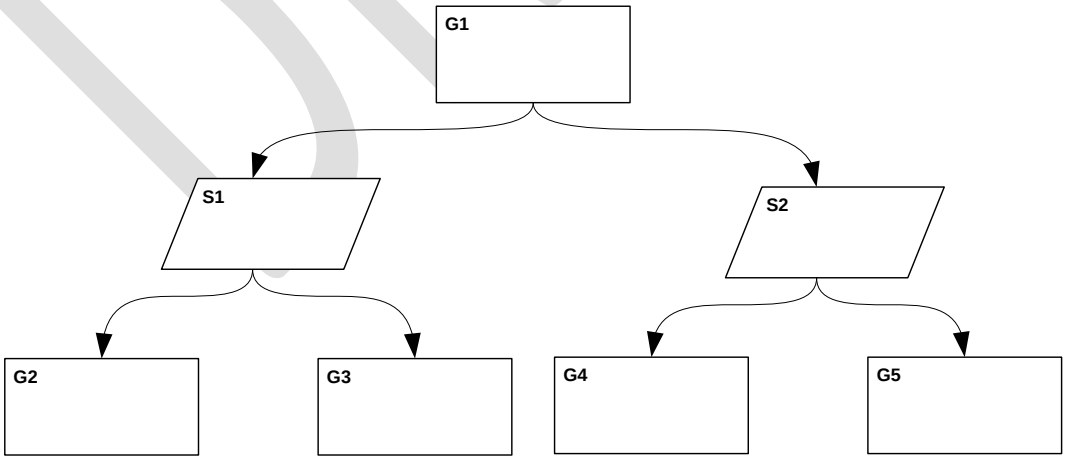


Figure 8: Multiple Strategies

Figure 9 represents the use of a reference to an evidence item to support a claim.

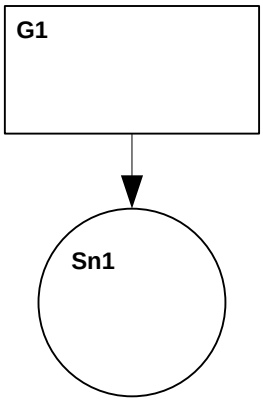


Figure 9: Providing Solutions

This structure represents an assertion that the statement made in the goal (G1) is established as true by the existence of the evidence referred to in the solution (Sn1). As with the use of multiple argument approaches to support a claim demonstrated in Figure 8, 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 should be presented in the goal structure. Figure 10 signifies an assertion that the statement presented in G1 is established as true by the existence of the evidence referenced by Solutions Sn1 and Sn2.

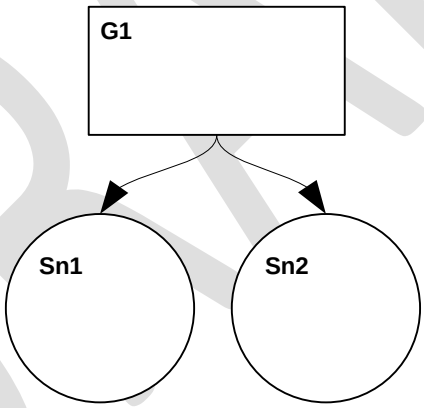


Figure 10: Multiple Solutions

When evaluating the support provided to a particular claim by lower-level claims, it is necessary to consider the context in which the claims are made. Figure 11 shows the addition of context to a goal. The context is used to declare supplementary information related to the claim made by Goal G1.

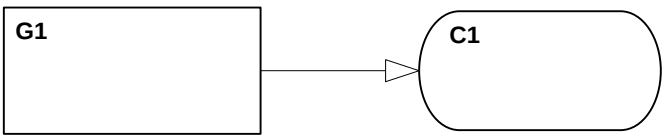


Figure 11: Adding a Context to a Goal

Contexts may be used to provide a definition or explanation of some term or terms used in the claim, or to 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.

An assumption applied to a goal declares an assumption made in stating the claim. Figure 12 illustrates the application of an assumption to a goal:

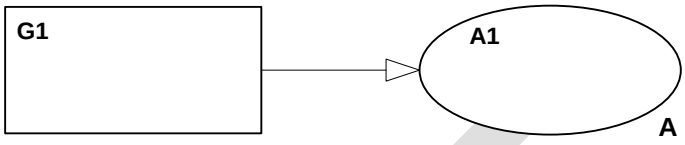


Figure 12: Adding an Assumption to a Goal

An assumption is an axiomatic claim: its truth is asserted as self-evident and does not require the support of a solution. The scope of an assumption is the entire argument. Having been stated once, there is no need to repeat or re-invoke it elsewhere. This does not preclude its being repeated or re-linked where this aids understanding.

Figure 13 shows the provision 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 selection or phrasing. The scope of a justification is limited to the element to which it is attached. Should the same justification be required elsewhere in the argument, it will need to be re-stated or re-linked.

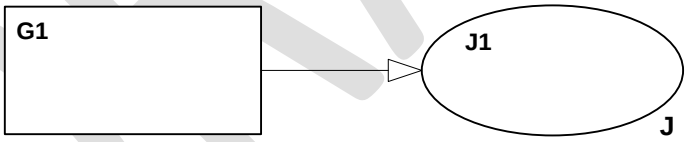


Figure 13: Adding a Justification to a Goal

A context may also be applied to a strategy to declare supplementary information related to explanation provided in the strategy or to provide a definition or explanation of terms used in the strategy. Figure 14 shows the addition of context to a strategy.

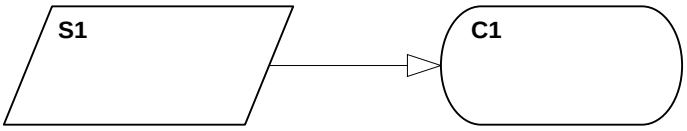


Figure 14: Adding a Context to a Strategy

As before, 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.

An assumption applied to a strategy declares an assumption in how the sub-goals support the parent goal. Figure 15 illustrates the application of an assumption to a strategy.

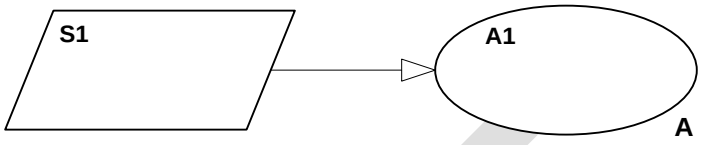


Figure 15: Adding an Assumption to a Strategy

A justification can also be applied to a strategy, to provide rationale for the selection or phasing of the argument strategy. The justification can supplement the strategy's explanation of support provided to the parent goal, by providing further explanation as to why the strategy is appropriate. Figure 16 shows the addition of a justification to a GSN strategy:

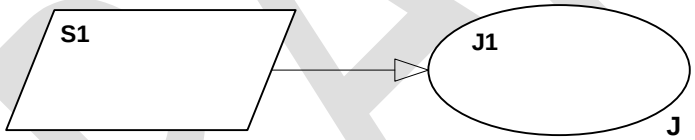


Figure 16: Adding a Justification to a Strategy

The scope of a justification is limited to the element to which it is attached. Should the same justification be required elsewhere in the argument, it will need to be re-stated or re-linked.

2.4 The Language of Goal Structures

The language used in GSN goal structures must reflect the logical relationships between elements established in, and enforced by, the symbology. A series of simple rules governs the grammatical structure of statements used to express particular elements of a GSN argument, and thus helps to preserve the logical structure of the argument.

GSN goals capture the propositions used in the argument (i.e. premises and conclusions). They should be expressed as <noun-phrase><verb-phrase> structures. The noun-phrase identifies the subject of the goal – i.e. the thing with

which the statement is concerned. The verb-phrase is used to define a predicate – it serves to make some assertion about the subject.

It is important to state goals atomically, that is to ensure that each goal 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 goals should be used, to ensure that the logical structure of the argument can be expressed clearly.

GSN strategies record the approach adopted in structuring the argument. Strategies are inserted between goals at two levels of abstraction, to explain how the top-level goal is addressed by the aggregation of the goals presented at the lower level. Strategies should not themselves form a necessary part of the argument – it should be possible to remove all strategy nodes from an argument without affecting the logical flow of the claims being made. Without them, an argument may be more difficult to follow, but it should remain in essential terms the same argument. Strategies are noun-phrase descriptions of the argument approach. Strategies should contain a brief description of the argument approach, introduced by a phrase such as “Argument by appeal to …”, “Argument by …”, “Argument across …”.

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

GSN contexts may be of two kinds. Where a context is a reference to an artefact of some kind, which informs the reasoning step, the context should be expressed as a noun-phrase. Contexts should be atomic (one artefact: one context). Where a context draws attention to explanatory contextual information (such as the definition of some term), this information can be stated briefly using complete sentences. The most effective sentence form to use is likely to be a <noun-phrase><verb-phrase> structure.

GSN assumptions and justifications provide additional information necessary for the correct understanding of the argument. Information should be stated as fully as necessary, using propositions in the form of complete sentences. Assumptions should be atomic.

ANNEXES TO PART 1

A1 EXTENSIONS TO GSN TO SUPPORT ARGUMENT PATTERNS

A1.1 Introductory

GSN can be used to articulate a specific safety argument. However, in order to generalise the specific details of a safety argument and represent patterns of argument rather than merely argument instances, GSN has been extended to support abstraction. Two forms of abstraction are supported:

- **Structural Abstraction**, which allows the generalisation of a relationship which exists between two specific instances 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.

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 safety claim. Entity abstraction allows generalisation (or postponement of detail) of an element in the argument structure. For example, for a particular failure rate goal, 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”.

A1.2 Structural Abstraction in GSN

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.

Figure 17 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 entity relate to another entity.

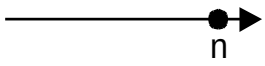
	A solid ball is the symbol for many (meaning zero or more). *The label next to the ball indicates the cardinality of the relationship.*
	A hollow ball indicates "optional" (meaning zero or one).

Figure 17: GSN Multiplicity Extensions (for Structural Abstraction)

The extension to GSN shown in Figure 18 enables the representation of structural options using the notation. This symbol is defined for use over all existing GSN relation types. Option can be used to denote possible alternatives in satisfying a relationship. It can represent 1-of-n and m-of-n selection.

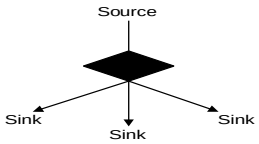
	1 source has three possible sinks Multiplicity relations can be combined with optionality relations. Placing multiplicity symbols prior to the 'option' vertex (solid diamond) describes a multiplicity over all the optional relations. Placing a multiplicity symbol on individual optional relations (i.e. just prior to the sink) describes a multiplicity over that relation only. It is useful to provide an annotation (next to the optionality symbol) denoting the nature of the choice to be made – e.g. "1 out of n" or "2 out of 3".

Figure 18: GSN Optionality Extensions (for Structural Abstraction)

A1.3 Entity Abstraction in GSN

Figure 19 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.

 **Undeveloped Entity**	This annotation denotes that the attached entity requires further development, i.e. at some later stage the entity needs to be (hierarchically) decomposed and further supported by sub-entities. Unlike uninstantiated elements, undeveloped elements are **not** replaced, they are further elaborated **in** the goal structure, i.e. as with undeveloped events in conventional Fault Tree Notation. This annotation can only be applied to goals and strategies.
 **Undeveloped and Uninstantiated Entity**	This annotation denotes that the attached entity requires both further development and instantiation.

Figure 19: GSN Extensions for Entity Abstraction

B1 MODULAR EXTENSIONS TO GSN

B1.1 Introductory

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

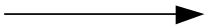
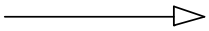
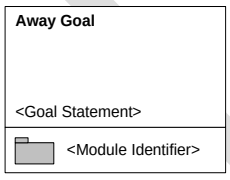
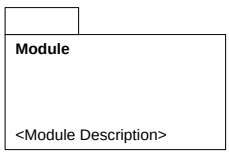
The following symbols are used in addition to the core GSN notation:

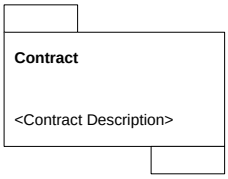
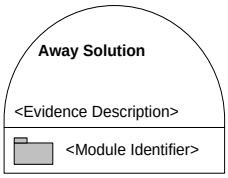
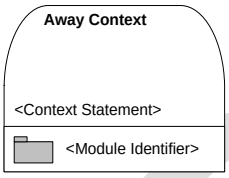
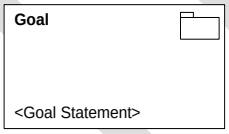
- Away Goal
- Module
- Contract
- Away Solution
- Away Context

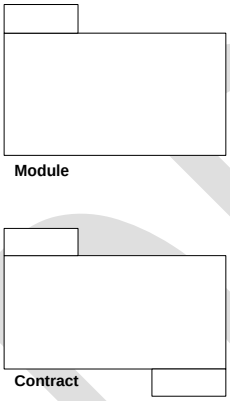
The concept of a 'module view' is also introduced. This uses a subset of the extended notation symbols to provide an abstract view of the argument structure.

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

The rest of this section provides the definition and rendering of these symbols and relationships. The meanings of structures combining these elements are further explained in Section B1.3.

Extensions to Core notation : Additional Permitted Connections	
	**Supported by**, In addition to the permitted connections defined in the core GSN definition, in 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’.
	**In Context of** In addition to the permitted connections defined in the core GSN definition, in 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.
Extensions to Core notation : New Entity symbols	
	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**, 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**, 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 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 artefact.
 Public Indicator symbol  Example of use (goal)	**Public** Indicator, rendered as a miniature module symbol and superimposed within (top right) a goal, solution or context symbol. 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 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 entity may be updated to replace this annotation with support from a named contract module, or may be left as is, with the necessary support defined in a higher level argument abstraction. This annotation can only be applied to Goal entities, and can be used in conjunction with 'To be instantiated' annotation, but is mutually exclusive with the 'To be developed' annotation.
Module View Subset	
	**Module and Contract** symbols are used in Module view to represent the module of argument without displaying the content of the argument. The arguments represented by these symbols are not necessarily captured in GSN.
	**Supported by** and **In Context of**, when used in the module view can represent one or more support/context relationship(s) between the entities within the modules.

B1.3 Notation Interpretation

Intra-Module Notation

The core GSN symbols defined in Section 1.2 and B1.2 above are intended to be combined to represent logical structures. The notation interpretation for core entities within modular extension is unchanged. Away goals, solutions and context are used in place of their core entity 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 module, rather decomposition needs to occur within the referenced module.

Arguments supported by another module can be indicated in a number of ways. Figure 20a illustrates a firm relationship that 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 entities in addition to the away goal.

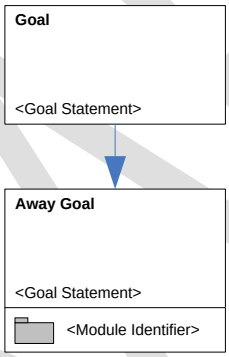


Figure 20a: use of 'Away Goals'

Figure 20b 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.

An alternative approach is illustrated in figure 20c. This has the same meaning, except that 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.

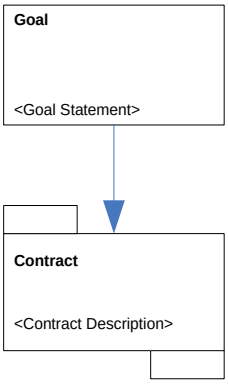


Figure 20b: use of contract

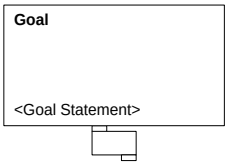


Figure 20c: use of unspecified contract

Where a Module is shown in support of a parent goal as illustrated in Figure 21 below, this signifies that the parent goal is supported by the entire argument made in the referenced module.

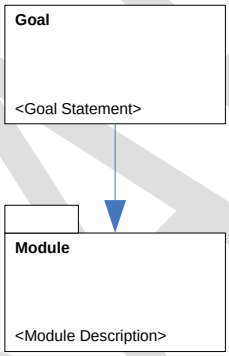


Figure 21: use of Module

There may be occasions when a goal or strategy requires fuller justification than can be provided within the confines of a normal GSN justification (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 22:

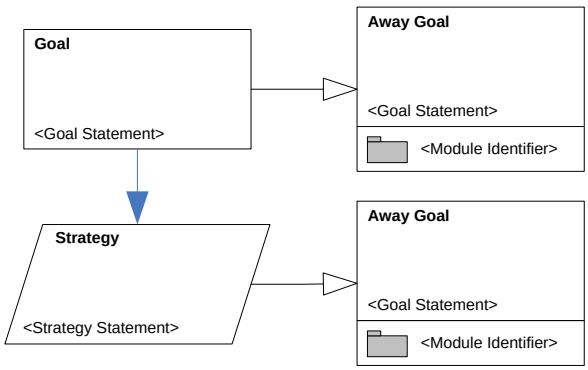


Figure 22: use of 'Away Goals' to replace Justification

Inter-Module Notation (Module View)

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

Figure 23a shows a 'supported by' relationship between modules. The relationship indicates that there exists one or more goal and/or strategy within the module 1 which is supported by one or more goal(s) and/or evidence entities within module 2, and similarly for modules 1 and 3. There is no inference that the support provided by module 2 and 3 are necessarily supporting the same goal in the in the module 1.

Unlike with goals, there is no implied hierarchy between modules. It is entirely permissible for a module to both provide support, and be supported by another module, provided that this does not create circularity within the overall argument.

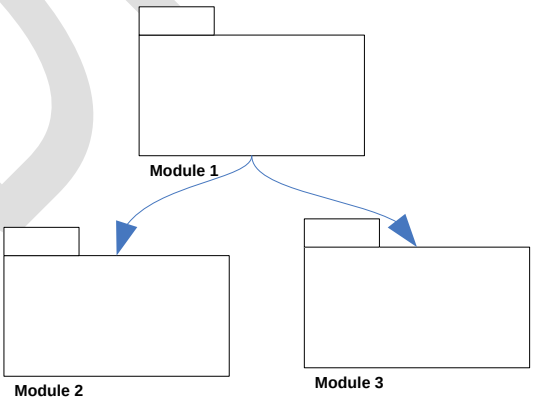


Figure 23a: 'Supported By' Relationship between Modules

Contract modules can be used in the support relationship between modules to aid decoupling as shown in Figure 23b. 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.

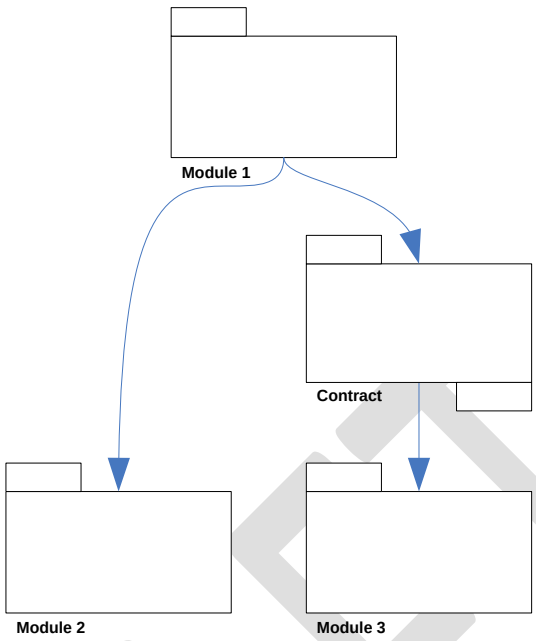


Figure 23b: 'Supported By' Relationship between Modules

The 'in context of' relationship between the two modules in Figure 24 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.

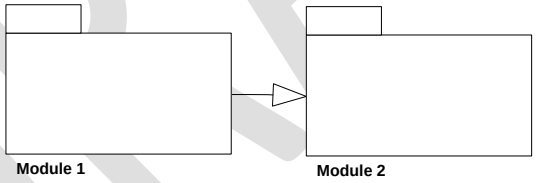


Figure 24: 'In Context of' Relationship between Modules

B1.3 The Language of Goal Structures

The modular extensions to GSN introduce a few additional language elements.

Modules need to be unambiguously identified, and therefore carry a module identifier. This identifier is used in away-goal, away-solution, away-context and module entities. The module identifier must uniquely identify a module within scope of the overall argument framework. For clarity of the argument, the module entity should carry a description of the nature of the argument contained within the module. The module description should be expressed as a noun-phrase.

The statement in away-goal, away-solution and away-context entities should exactly match that in their referenced module counterparts.

3 GUIDANCE ON THE DEVELOPMENT AND EVALUATION OF GOAL STRUCTURES

3.1 Introductory

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

- **Clarity** of the documented argument – individual claims and references must be easily understandable, and the logical flow of the argument must be clear.
- **Intelligibility** of the documented argument – writer and reader must share an understanding of the claims being made. Where necessary, the writer should provide details of the context in which the argument is being put forward.
- **Defensibility** of the documented argument – where appropriate, the writer must provide rationale for the argument approach he has adopted and its appropriateness in the context in which it is proposed.
- **Validity** of the documented argument – the documented argument should accurately reflect the state of the evidence and reasoning at the time of writing.

This section of the standard is intended to provide pragmatic guidance for the author, to help him produce clear, intelligible and defensible argument structures using GSN. Although 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. This variety of approaches is reflected in the guidance given in this section: Section 2.2 describes top-down approaches to argument development, while Section 2.3 looks at bottom-up approaches. Section 2.4 addresses the derivation of goal-structures from existing textual arguments. 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 safety arguments, while Section 2.8 examines the various roles goal structures can perform in the project lifecycle.

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

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 safety argument for a fictional automated press system, is used to clarify concepts introduced during the discussion.

3.2.1 Overview

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.

Figure 25 illustrates this six-step process, which is recursive. Given a 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 goal immediately through reference to some basic evidence (step 6). More commonly, however, it will be necessary to identify some intermediate goals, to refine the argument, incrementally, to a level of detail at which the goal 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).

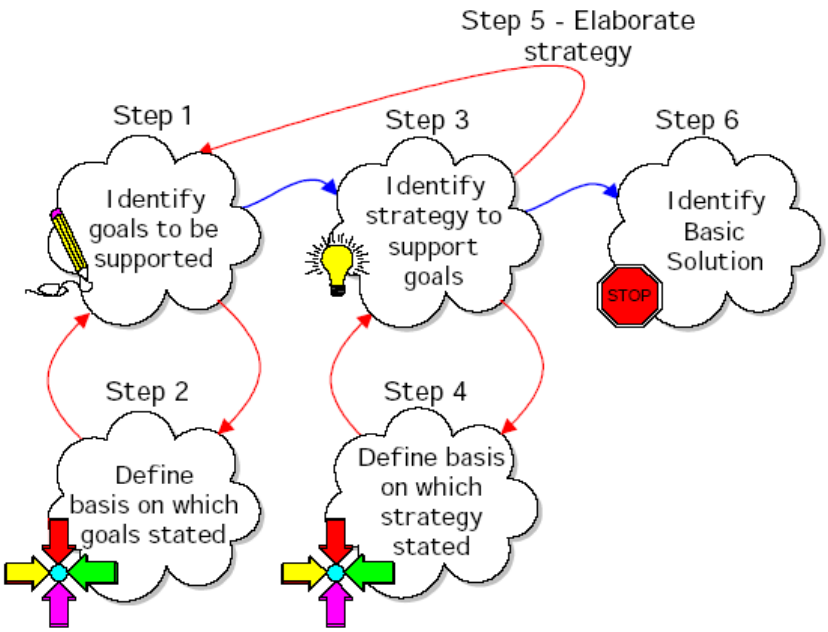


Figure 25: Six-Step Process for Developing Goal Structures

3.2.2 Step 1: Goal-Identification

The objective of this step is to identify the top goal(s) of the structure, the principal statement(s) that the remainder of the argument should support. It is important that 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 top goal 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 goal from that fundamental objective. Figure 26 introduces the top goal of a running example used to illustrate the top-down development of a goal structure in this section:

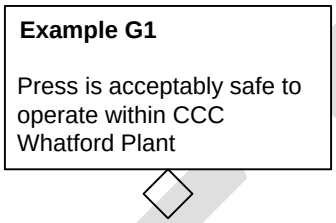


Figure 26: Top Goal of Running Example

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

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 goal(s) identified in step 1 are put forward. There are three key aspects to this activity:

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

GSN Contexts are used to refer to system information, artefacts or processes (see Section 1.3 above). Figure 27 illustrates the association of context with a goal, to clarify concepts introduced in the claim:

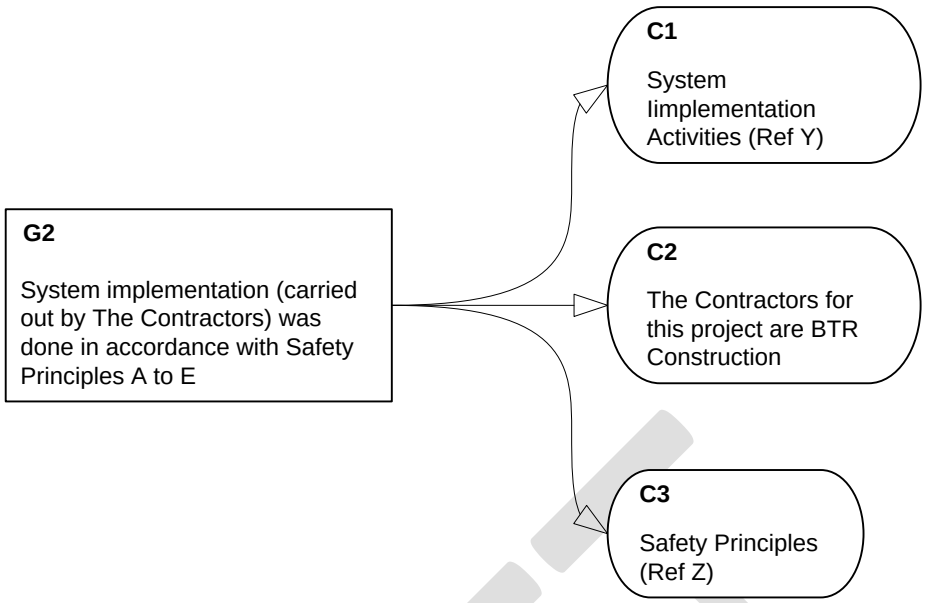


Figure 27: Association of Additional Contextual Information

In Figure 27, goal G2 introduces three terms which potentially require clarification for the reader: “system implementation”, “the contractors” and “safety principles A through 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.

Note that, as discussed in Section 1.3 above, contextual information associated with a 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.

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 claim “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. 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

Figure 28 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 captured in Goal G1:

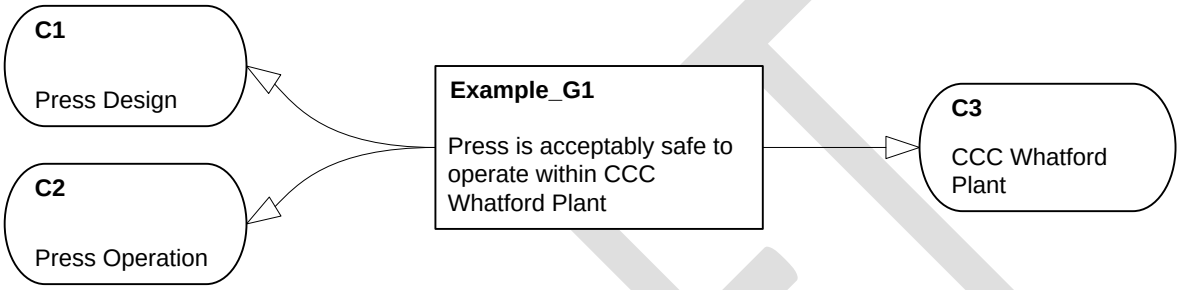


Figure 28: Example with Contextual Explanation

In Figure 28, the terms 'press', 'operate' and 'CCC Whatford Plant' have been drawn out into explicit GSN contexts, 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.

3.2.4 Step 3: Identification of Strategy

Having identified and expressed a goal and explicitly stated the context in which it is stated, the author's next task is to work out how the goal 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?

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 goal. 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 29 illustrates this approach:

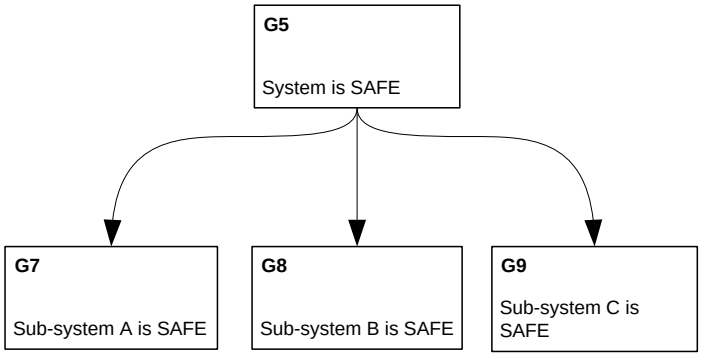


Figure 29: Divide and Conquer Goal-Decomposition

Another common approach is to attempt to re-state the original goal 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 31 illustrates this approach:

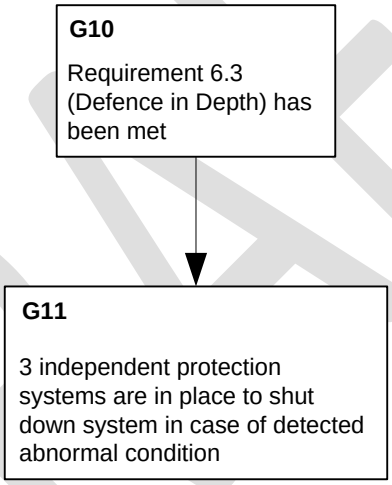


Figure 30: Interpretation, or Particularisation, of a Goal

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

Figure 31 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.

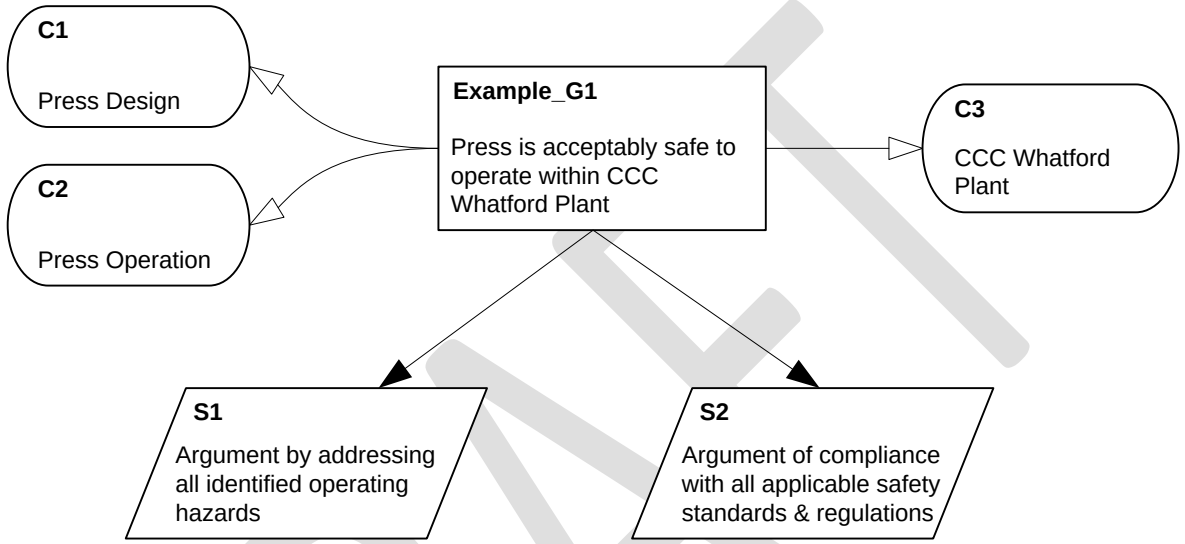


Figure 31: Example with top-level strategies

3.2.5 Step 4: Definition of the Basis on which the Strategy is Stated

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, goals 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 31 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.

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 a solution to 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

Continuing the development of the goal structure from Figures 31 and 32, Figure 32 shows the contextual information necessary to clarify Strategies S1 and S2:

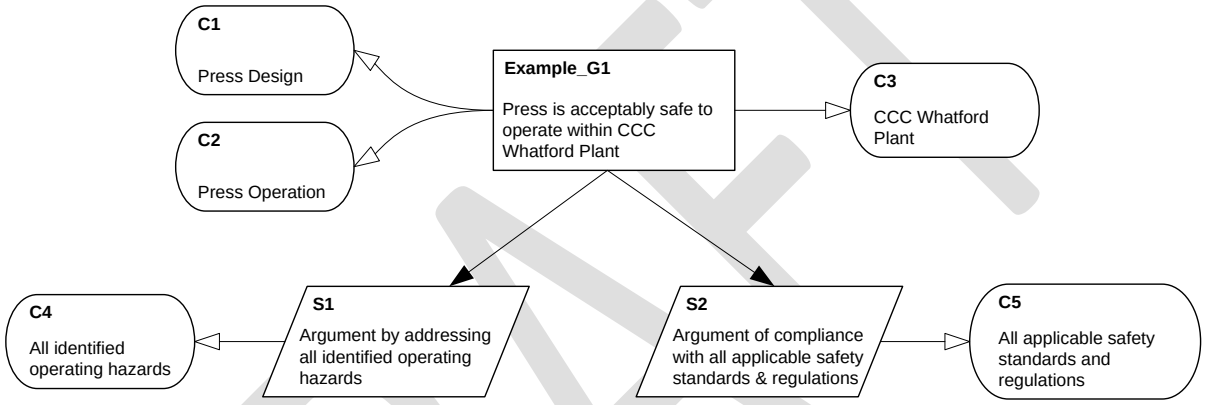


Figure 32: Example with contextual evidence to clarify strategies

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

3.2.6 Step 5: Elaborate Strategy

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 represented by 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 going to be made concerning all of a system's constituent sub-systems, appropriate goals are put forward 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.

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 node. It is important to realise that, logically, there is always a strategy underlying the argument’s construction.

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.

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

Figure 33 shows the elaboration of the strategies defined in Figures 32 and 33. 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, 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).

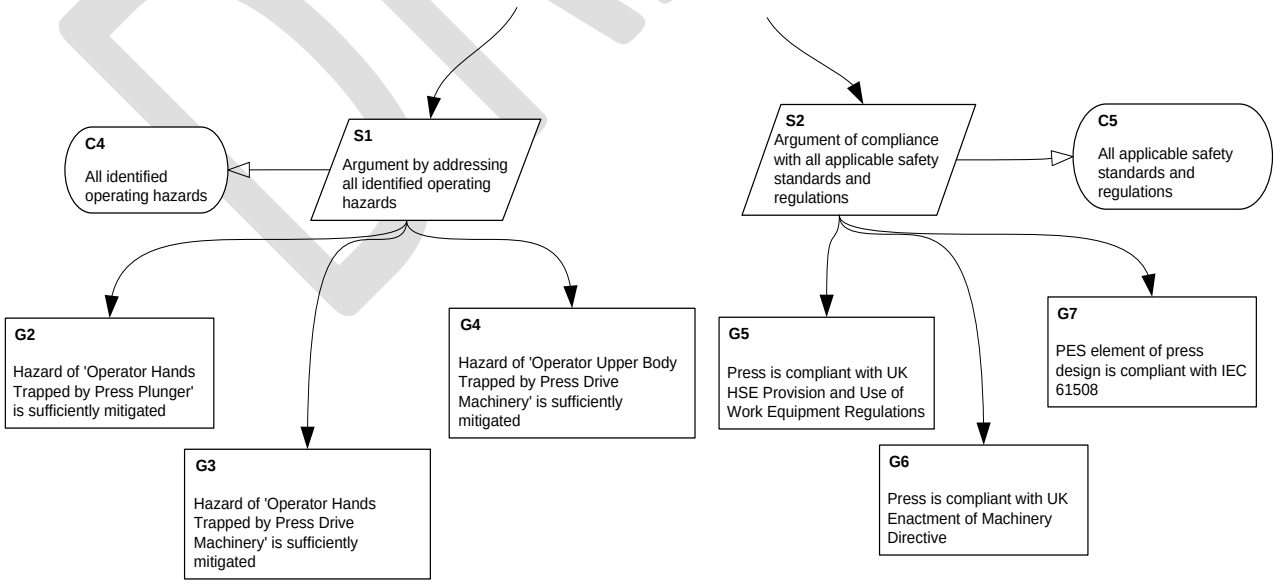


Figure 33: Elaboration of Strategies

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).

3.2.7 Step 6: Identify Solutions

Eventually, goals 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 external evidence. In GSN, a solution is added to support the goal (see Section 1.3 above). Figure 34 shows the fragment of goal structure developed to support Goal G3, which was derived from the application of Strategy S1 in Step 5 (Figure 34). The author has decided that no further decomposition of the claim "Motor / clutch/ drive belts surrounded with safety cage" is required, and that this claim will be validated by reference to the design of the press.

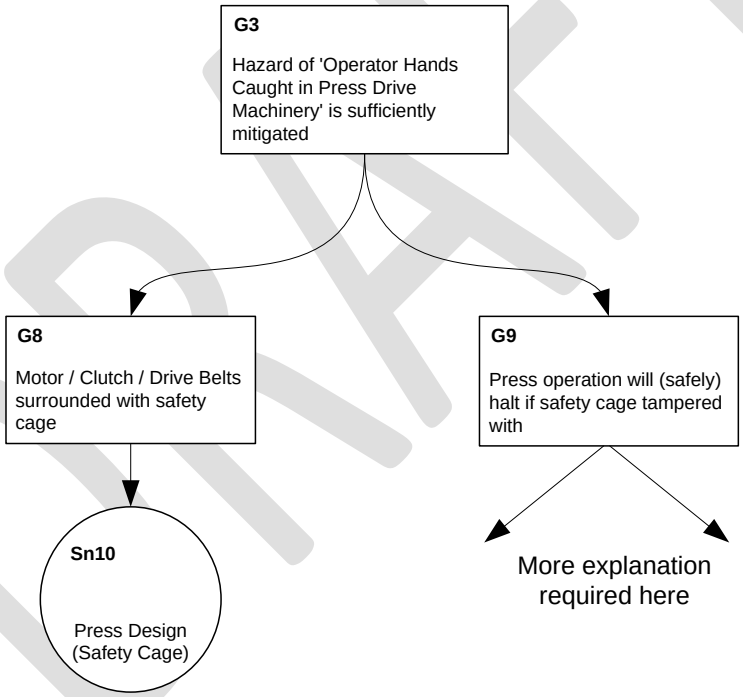


Figure 34: Reference to evidential support

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.

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

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.

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.

3.2.8 What if we can't close out the argument?

A frequent problem in top-down argument development is that the author gets some way in the decomposition of some goal or claim and then realises that there is insufficient evidence available to enable him to 'close out' the claim. Either the evidence he requires is missing, or, as is more frequently the case, the evidence he has does not 'cover' the lowest-level claim he wishes to make 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 he would in the bottom-up argument development approaches described in Section 2.3 below), and establish what claim it will allow him to make. The claim immediately above the GSN solution 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.

3.3 Developing Goal Structures Bottom-Up: Working from Available Evidence

3.3.1 Introductory

It is sometimes necessary or useful to build a GSN safety 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 cases where an existing safety case must be updated or improved. Production of a safety or 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.

Adapting Kelly's six steps (Section 2.2) for top-down GSN development, the following process can be used to develop a goal structure bottom-up:

1. Identify evidence to present as solutions
2. Infer "evidence assertion" goals to be directly supported by these solutions
3. Derive higher-level sub-goals that are supported by the evidence assertions
4. Describe how each layer of sub-goals satisfy their 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 known top goal or set of sub-goals

Figure 35 shows these steps graphically:

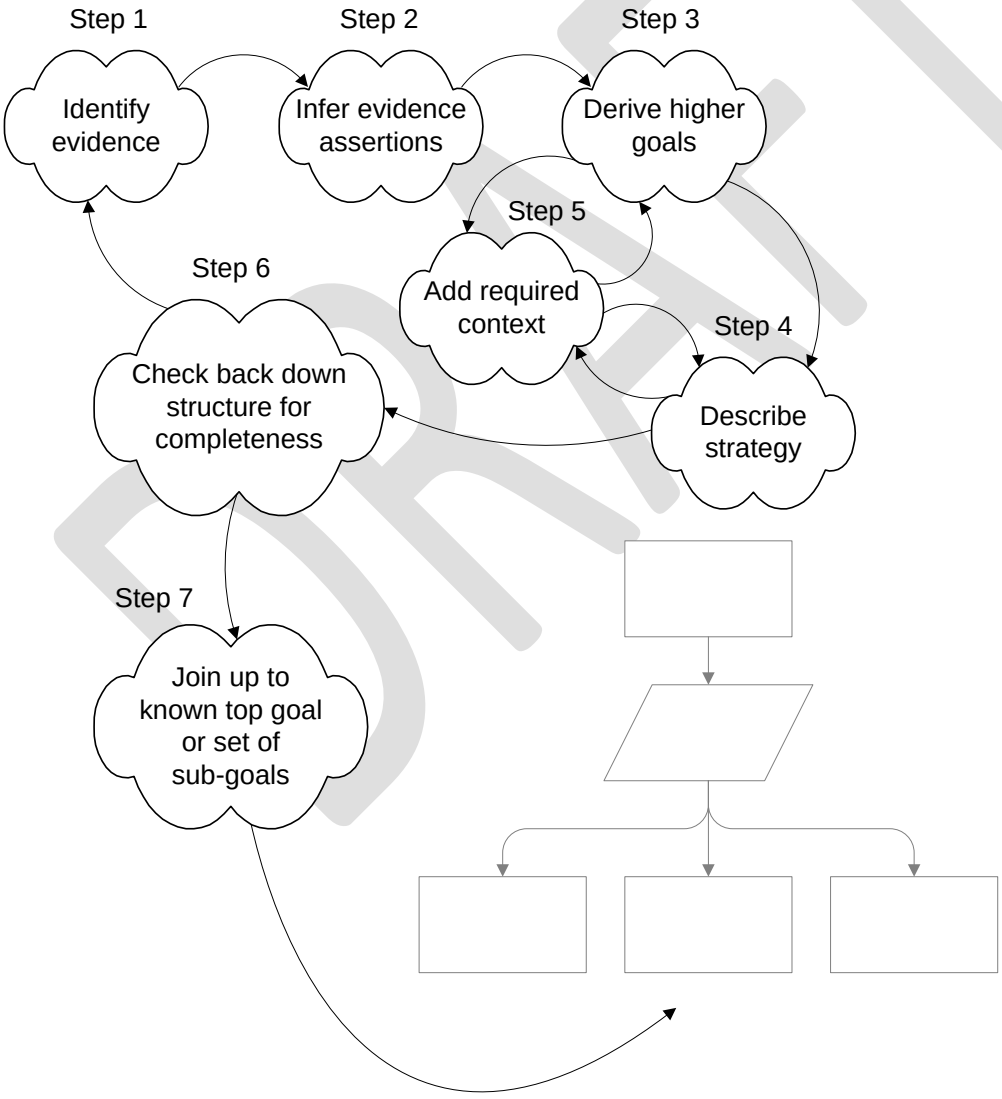


Figure 35: Bottom-up Process for Developing Goal Structures

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. geographical layout or interlocks, rather than having been developed to a specific process.

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 GSN arguments.

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.

3.3.2 Bottom-Up Step 1: Identify Relevant Evidence

In developing a GSN safety 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 Effect Criticality Analysis (FMECA), shown in Figure 36:

Figure 36: Typical Solutions Derived from Evidence

Having created solutions from analysis such as Fault Tree Analysis, the author should consider what the 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 solutions will support (see bottom-up Step 2).

3.3.3 Bottom-Up Step 2: Infer “evidence assertion” goals

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 reveal this claim, which is

represented as a bottom-level 'evidence assertion' goal in the safety argument (see Section 2.2.7), inferred directly from the available evidence. They differ from higher goals in that the subject is the evidence rather than the system property in question. Figure 37 demonstrates the inference of evidence assertion sub-goals directly from solutions:

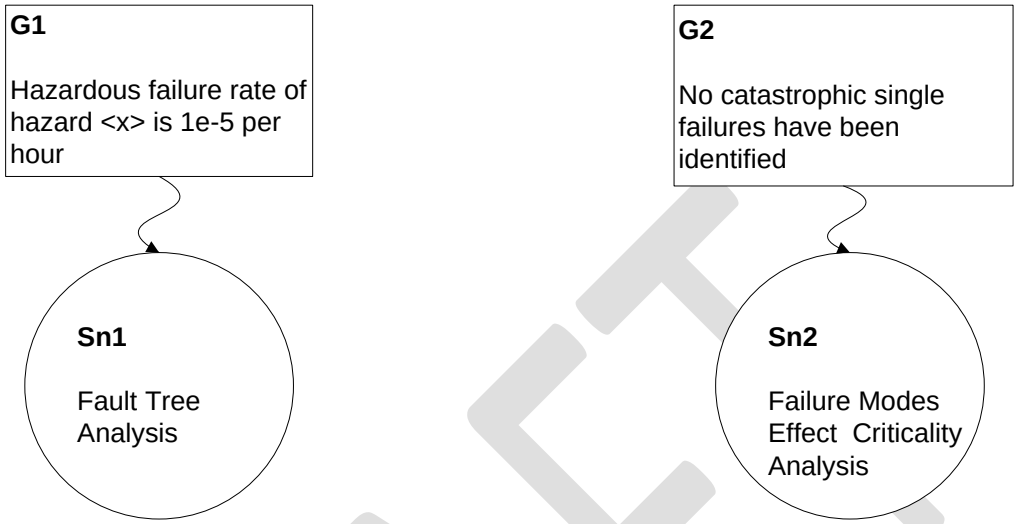


Figure 37: Evidence Assertion Sub-Goals Inferred from Solutions

The goals captured using this approach can then be built into the safety argument using the process described in bottom-up Step 3 below.

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 38 illustrates this approach:

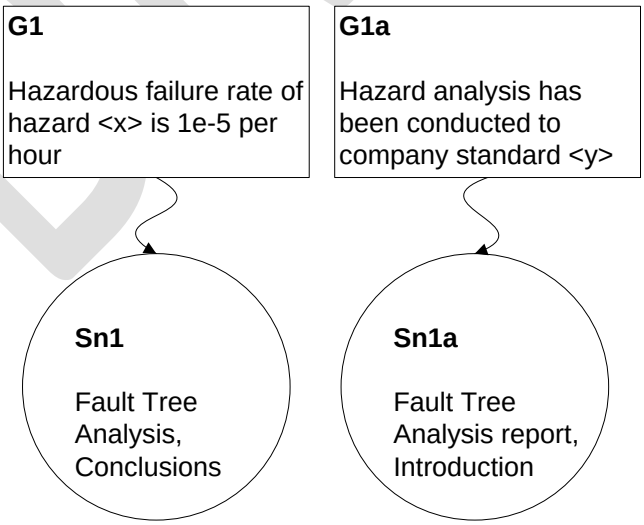


Figure 38: Multiple Evidence Assertion Sub-Goals Inferred Similar Solutions

3.3.4 Bottom-Up Step 3: Adding Higher Sub-Goals

Having constructed the bottom of the goal structure as a series of solutions (represented by the available evidence) and evidence assertions 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 higher-level claim (see Introduction). Figure 39 illustrates adding a higher-level sub-goal:

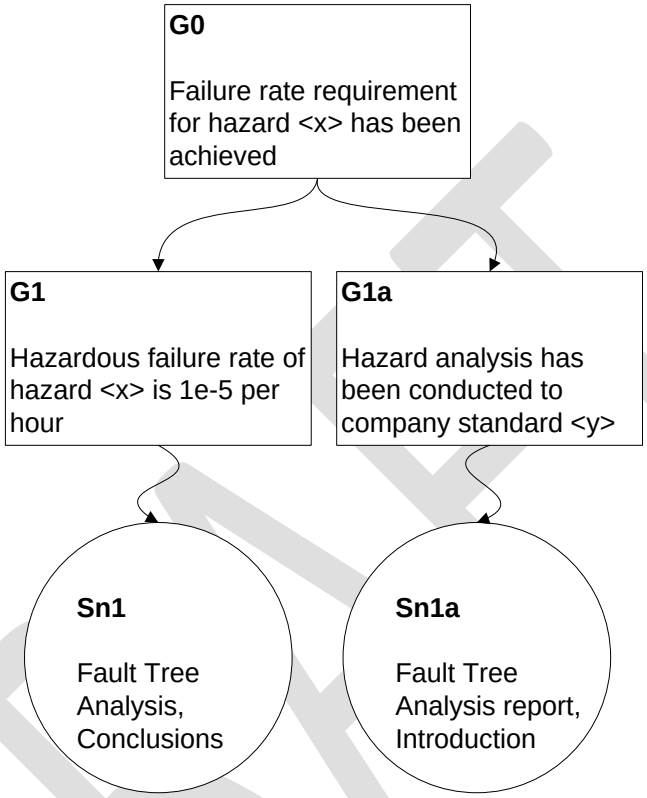


Figure 39: Adding a Higher-Level Sub-Goal

In considering how 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.

Goals should not be exclusively product-oriented – often, process evidence can be obtained from entities like the FTA used as an example earlier. 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.

Note that the evidence assertion and support solution of FMECA evidence has been omitted from this fragment – the same steps are required to complete that area of the argument. The author should not be pressured into manipulating evidence to fit under evidence assertions or goals that do not directly relate – allow the argument to develop naturally.

3.3.5 Bottom-Up Step 4: Describe Strategy for goal-decomposition

When deriving a higher-level goal that is supported by its sub-goals, it can be helpful to describe how those sub-goals satisfy the parent goal. Note that unlike the top-down process, the author will seldom have any choice about how the goal to sub-goal decomposition is achieved – hence the use of the term “describe” when applied to showing strategy. Figure 40 shows the addition of a strategy to implicitly describe the step between parent goal and sub-goals:

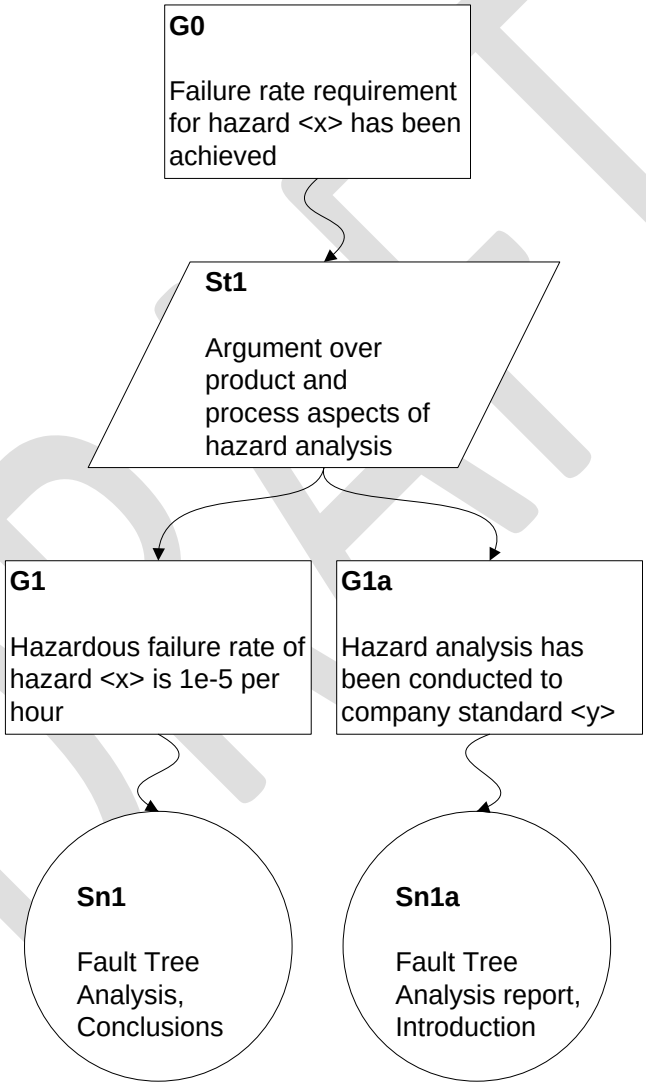


Figure 40: Describing the Strategy for Goal Decomposition

Should the statement of strategy be obvious, it is not always necessary to represent it as part of the goal structure. However, it is crucial that the author understands what strategy has been adopted in order to complete the following steps.

3.3.6 Bottom-Up Step 5: Adding Contextual Information

The creation of a goal structure from existing evidence may have elicited contextual information, including assumptions, definitions and references. Figure 41 shows the addition of contexts to the parent goal:

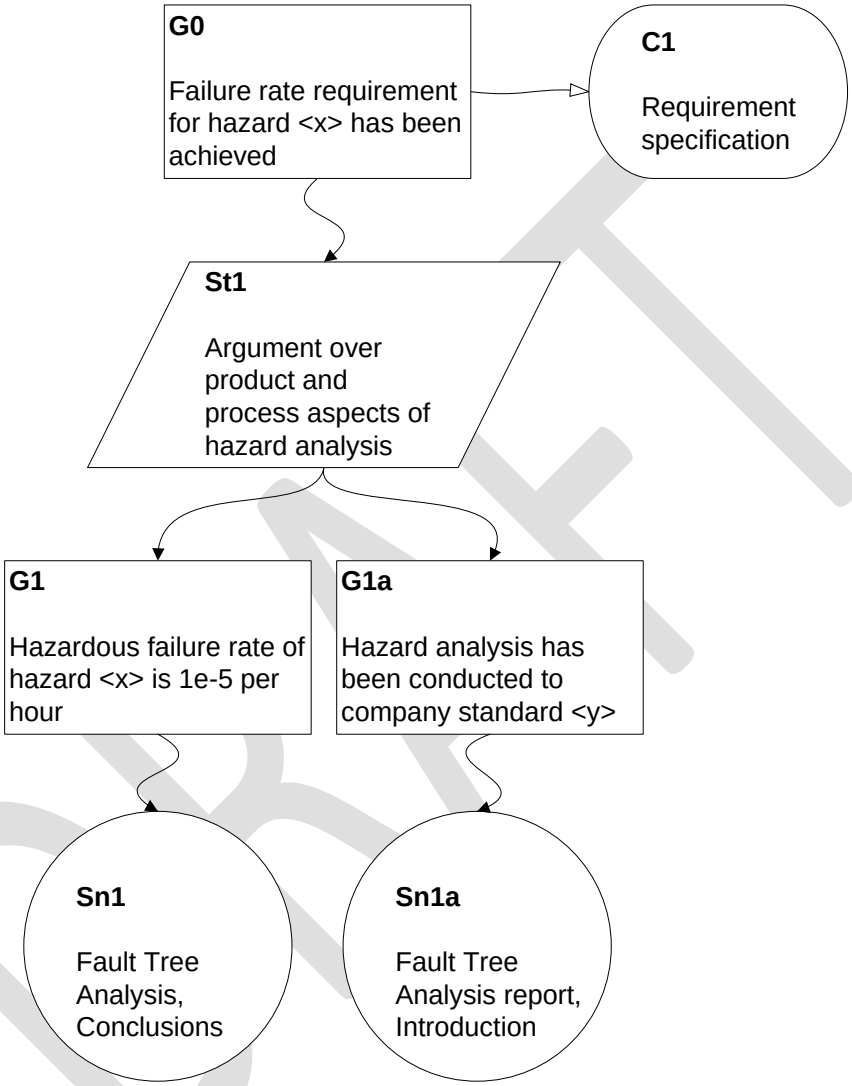


Figure 41: Describing the strategy for goal decomposition

For the example of FTA used earlier, 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, thus providing scope for the bottom level of evidence assertions made in the argument.
- System usage assumptions 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 bottom-up, these considerations can be useful to ensure completeness.

3.3.7 Bottom-Up Step 6: Check back down the goal structure

At each step of creating parent goals, the author should re-examine (top-down) the supporting sub-goals to check for adequate support of the parent goal. This exercise should also extract the strategies used to make the inference between sub-goals and parent goal

However the high level goal structure is arrived at, it is recommended that at each step up the argument structure, a reflective look back down is made. This should consider whether the supporting goals provide sufficient coverage and support to the newly-created parent, and whether any assumptions or other context has been relied upon to mane the inference step. The results of this evaluative reflection may indicate that other supporting goals. Solutions or context needs to be introduced, or that the claim needs to be rephrased.

In the example goal structure used in the diagrams above, one output of this “check back down” step might be to identify the requirement for operator competence to conduct FTA or demonstration of absence of common causes.

3.3.8 Bottom-Up Step 7: Incorporate bottom-up goal structure into higher (top-down) argument

As already stated, 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.

As the goal structure is developed from the existing evidence, the author should keep in mind where the argument is “aiming” i.e. attempt to write it in such a way that it bridges the gap between a known argument claim higher up and the existing evidence. Figure 42 illustrates this connection:

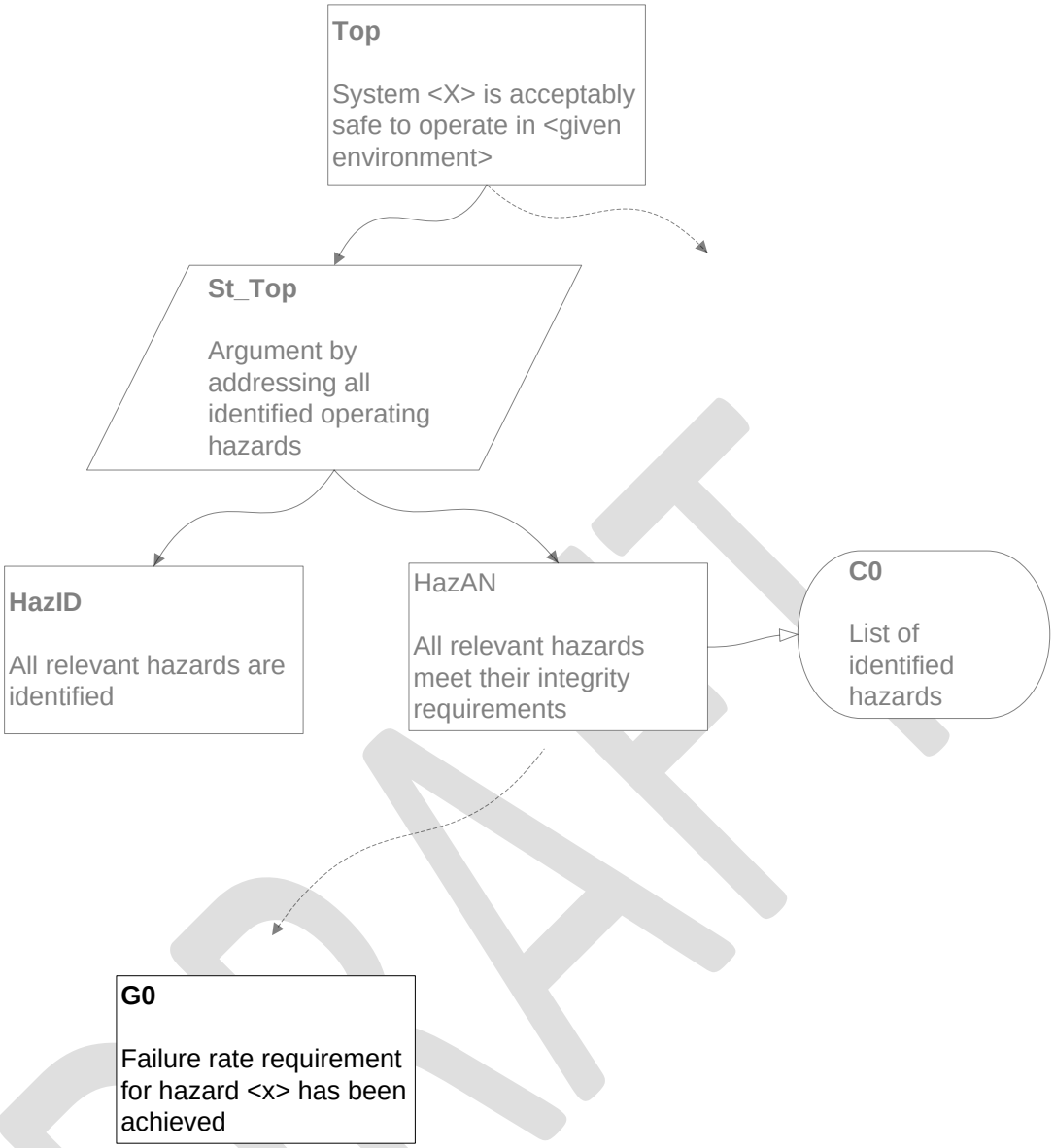


Figure 42: Joining the Bottom-Up Goal Structure to a higher fragment

3.3.9 What if I can't convince myself?

There will be circumstances in which, on assessing the argument built from the evidence available, the author realises that the evidence is inadequate to support the created claims with the required degree of confidence. The evidence might, for example, be incomplete, or might relate to a different version of the system from that addressed by his 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 is honest about the limitations of the evidence he has, and scopes his claims accordingly. Where possible, claims which are potentially undermined by shortcomings in one evidence artefact should appeal to more than one solution for support.

3.4 Understanding Existing Arguments

Text to be supplied.

3.5 Avoidance of Common Errors in Creating Goal Structures: (1) Language Issues

The guidance presented in Sections 2.5 and 2.6 is based on 'real-world' experience of goal-structure development. 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 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.

3.5.1 The 'Essay in the Box'

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 node. 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. In general, the textual element of GSN arguments should be kept as brief as possible, though strategies, justifications, assumptions and textual definitions should be stated in 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** – GSN goals contexts and solutions 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 often indicates that the goal contains multiple claims, as does the existence of more than one noun-phrase preceding a single verb-phrase. Where contexts or solutions 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 content placed in GSN nodes reflect the logical function for which the GSN

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 this 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 artefact provides a direct solution.

- **Allow contexts to act as references**. As defined in Section 1.3 above, contexts and solutions in GSN should provide references to artefacts stored elsewhere. A single noun-phrase (perhaps accompanied by a further reference to the location) should be sufficient to identify these artefacts. It is not necessary to summarise the content of the artefact in the GSN node.

3.5.2 Ambiguity

'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. 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" means something rather different to a speaker of US English (who would interpret 'momentarily' to mean "in a moment, presently") than to a speaker of British English, who would expect the light to flash only once, for a short time.

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 grammatical 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 limit.

3.5.3 Vagueness

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 GSN claim, 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'.

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 ask himself whether so strong a claim is in fact valid. 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.

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 safety argument.

3.5.4 Oversimplification

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.

3.6 Avoidance of Common Errors in Creating Goal Structures: (2) Structural Issues

3.6.1 Jumping Ahead

One of the potential dangers associated with defining the top goal of an argument is 'jumping ahead', i.e. stating a goal which supports the overall objective of the argument, rather than actually stating the objective itself. For example, the author of a safety 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 goal, the author should ask himself 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.

3.6.2 Erroneous use of Context

In GSN, contexts should not be used to refer to information which is intended to support the validity of a goal. Such information is evidence for the truth of the claim made in the goal, and as such should be represented using a GSN solution. Figure 43 illustrates this incorrect use of a GSN context to support a goal:

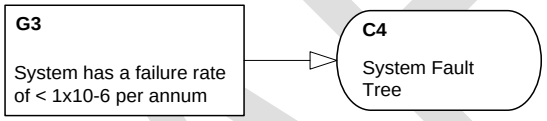


Figure 43: Incorrect use of Context (as a Solution)

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.

Context is sometimes used where a GSN assumption or justification may be more appropriate. In Figure 44, for example, the statement "System X has no Common Mode Failures" would be more appropriately rendered as an assumption than as a context:

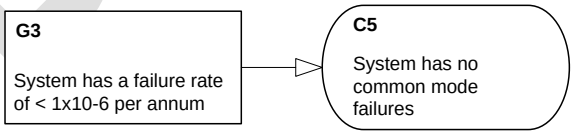


Figure 44: Incorrect use of Context (as an Assumption):

3.6.3 Erroneous use of Strategies

In GSN, Strategies are intended as a description of the argument approach which has been carried out to relate goals 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 the design.

Another common mistake is for 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. These claims can either be made explicitly as part of the strategy or can be implied. Claims contained in strategies, rather than in goals, cannot be properly supported by the subsequent goal structure, and will therefore remain undeveloped in the argument.

3.6.4 'Leaps of Faith'

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.

In arguments represented in 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 45, 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:

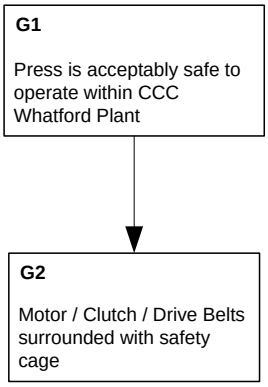


Figure 45: An Inductive Leap

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

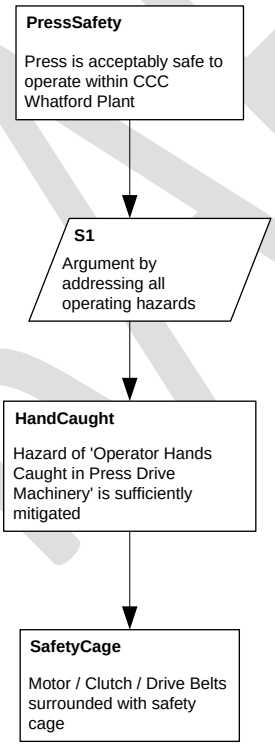


Figure 46: Intermediate Goal as a 'Stepping Stone'

Another common error is for the author to attempt to 'close out' a goal prematurely by direct reference to evidence in a way which will not be easily understood by the reader. For example, consider the Solution provided in Figure 47:

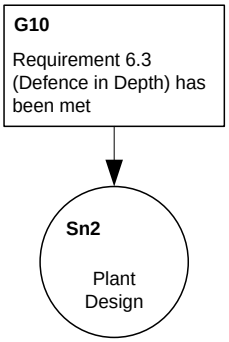


Figure 47: 'Jumping' to a Solution

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 Goal G10 can be inferred from 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.

3.7 Evaluating Goal Structures: A Step-by-Step Approach

3.7.1 Introductory

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, goal-structured argument cases are often subjective and have many stakeholders. This section provides a step-by-step approach to the review of goal structures and guidance on assessment of the level of assurance the argument provides.

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.

3.7.2 The Role of Review in the Lifecycle

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. However, in terms of risk to the project, staged review is a far less risky approach. If there are problems with the arguments and evidence being offered up, it is desirable that this be discovered as early as possible in the lifecycle.

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.

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?"

3.7.3 Problems Commonly Experienced in Reviews

A key difficulty reported by those regularly involved 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, of the argument.

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 of the reader. It will almost always 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.

3.7.4 A Staged Argument Review Process

Figure 48 illustrates a staged approach to the review of assurance case arguments, derived from [6].

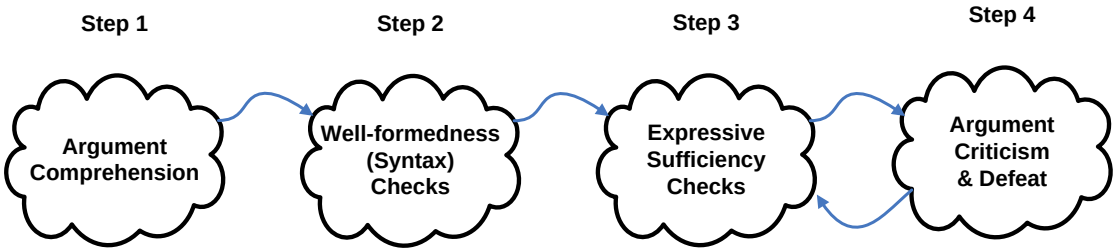


Figure 48: Staged Argument Review Process

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

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 does the former.

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 adequately (the concern of step 3).

The following sections describe the activities and concerns of each of the four steps of the review process:

3.7.4.1 Step 1: Argument Comprehension

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 captured 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 nodes 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.

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.

3.7.4.2 Step 2: Well-Formedness Checks

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.

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.

3.7.4.3 Step 3: Expressive Sufficiency Checks

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 node in GSN is to explain the relationship between a parent goal and 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.

Equally, it is possible to add references to contextual information in GSN wherever the meaning of a goal or strategy is unclear (See Section 1.3 above). In this review step, it may be necessary to demand further context to be defined before any further review should sensibly 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.

3.7.4.4 Step 4: Argument Criticism and Defeat

Safety 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? 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.

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".

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).

3.7.4.5 Auditing the Evidence

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.

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.

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.

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.*

Consideration of counter-evidence is one of the most difficult aspects of safety 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.

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.

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.

3.8 Goal-Structuring and the Project Lifecycle

3.8.1 Evolution of the Argument in step with a Project

It is generally recommended that an assurance argument or safety argument be developed early in a project. This allows for early visibility of the argument approach, and feedback into technical planning activities to ensure that any analysis and evidence production activities necessary to support the argument are in place. This provides a basis for later review of the argument, and for adjustments to be made to the argument strategies as necessary.

Even if the argument presented at an early stage in system development is high-level and incomplete, it can provide a framework for further review, development and refinement. Early feedback can save wasted effort through:

- Exposure to technical review – is the proposed argument approach feasible?
- Exposure to regulatory review – will the proposed argument approach be acceptable to the regulators?
- Exposure to stakeholder review – does the proposed argument approach provide what is required?
- Exposure to management review – have we provided the relevant evidence on time (or will we have done so?)

Development of a partial safety argument prior to system development can also have a role in informing the planning of safety activities, their requirements and timings. Some standards, including Defence Standard 00-56 (Issue 4) [8] support the

incremental updating of a safety argument in a series of Safety Case Reports or Assurance Case Reports issued at intervals throughout the lifecycle.

A preliminary safety argument is primarily intended to provide a structure for the claims concerning safety which will be made in the evolving safety or assurance case report. The preliminary argument also provides the development team with some assurance that the final Safety Case – populated with detail and evidence – will be acceptable. The preliminary argument is likely to take the form of a set of top-level goals supported by some outline strategies and justifications, and also the contextual basis for the goals and strategies presented. At this stage, it is unlikely that there will be much evidence available to the author, except for some backing evidence for the design and development process which may be used. This absence of evidence should not be seen as an obstacle to generation of the preliminary safety argument. Thought should be given, even at this early stage, as to what evidence would satisfy the argument's claims. GSN solutions can usefully be introduced into the preliminary safety argument as evidence obligations to be satisfied later in the process. It is useful to apply information abstraction techniques at this stage, to make the argument clear and free from unnecessary clutter relating to later phases.

Where a goal cannot be developed in full in an early-stage argument, an undeveloped goal can legitimately be presented (see Section 1.2 above). This is a useful approach to adopt when the design has not matured to a stage at which the likely argument approaches can be determined, for example. In cases where reuse may apply, GSN modular extensions can be deployed (see Annex B1).

An evolving safety argument can also be used as a tool to support the safety management of a project. Evaluating the status of the safety argument can facilitate monitoring of progress towards the achievement of safety requirements. Progress against a safety plan can be evaluated by examination of the population of evidence in the safety argument. Similarly, highlighting items where the evidence to support the argument is missing allows focus on activities to meet the plan.

Different domains will have their own development lifecycles, for example commissioning, pre-operational, operational, decommissioning. The safety argument can reflect this. This can be through the selection of appropriate development goals that relate to users of the phase. The lifecycle-related arguments are often associated with phased safety cases.

3.8.2 Retrospective Argument Construction

When developing a safety argument retrospectively, the author's main concern is to establish what "drives" the argument at each stage. Two principal approaches are possible; the argument can either be driven from the available evidence or derived

top-down from the top-level claim (see sections 2.2 and 2.3 above). Evidence-driven argumentation (section 2.3) is the simpler approach and a legitimate argument can be made, if the claims are fully supported by the evidence available. The weakness of this approach is that any limitations in the argument may not be readily identifiable. Top-down approaches (section 2.2) are more applicable if there is a requirement for the safety of the system to be demonstrated against current standards. The argument may produce evidence obligations which cannot be supported by the evidence available, i.e. an evidence gap. In this case, the argument will require revision to provide justification as to why the system can still be held to be acceptably safe, even with the gap in evidence produced from the original reasoning. The weakness of this approach is that it may take considerable effort to justify the gaps in what is otherwise an acceptable system.

3.8.3 Use of GSN in Review and Acceptance

GSN may be used as a tool to support the review and acceptance of the system. The argument produced in this phase is not a safety argument as such, but rather an assurance argument. For example, an Independent Safety Assessor may use GSN to construct an argument concerning the status and coverage of his assessment to support a top-level goal of the form “an independent safety assessment of System X has been completed”. The argument would use evidence relating to plans, processes and the product to justify the claim.

ANNEXES TO PART 2

A2 GUIDANCE ON PATTERN EXTENSIONS

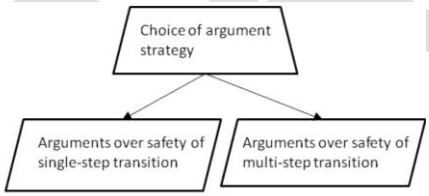
To be supplied

B2 GUIDANCE ON MODULAR EXTENSIONS

To be supplied

C2 OTHER EXTENSIONS TO GSN

From time to time, symbols 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 42 illustrates the symbols used, and explains the concepts they are intended to represent. A number of these symbols derive from the use of GSN in requirements capture and analysis. With the exception of the "choice of strategy" symbol, all of these symbols can be replaced by suitably-worded context symbols 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*

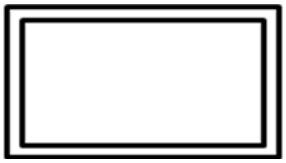
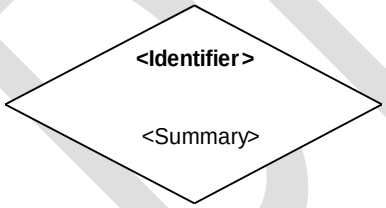
Constraint 	This is a form of context symbol which is used to indicate a constraint that might impact the way in which the goal to which it is attached can be supported. Example: *Source code of component not available for inspection*
Stakeholder 	This is a form of context symbol which is used to indicate 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 49: Other GSN Extensions

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.

Structured argument

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

Evidence

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

Claim

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

Assurance Case

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

REFERENCES

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

[2] ASAM-II ref

[3] S. Toulmin, *The Uses of Argument* (1958; second edition 2003)

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

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