A Pragmatic Approach to Reasoning about the Assurance of Safety Arguments

Rob Weaver, Jane Fenn and Tim Kelly
Department of Computer Science
The University of York, York YO10 5DD UK

BAE SYSTEMS
Warton Aerodrome, Preston, PR4 1AX
{rob.weaver, tim.kelly}@cs.york.ac.uk
jane.fenn@baesystems.com

Abstract

The development of safety critical systems is guided by standards. Many standards require the development of a safety case to demonstrate the acceptability of Safety Critical Systems. The safety case must provide confidence that the system is deemed safe enough to operate. For system components where it is not possible to quantify the associated risks (e.g. software), current standards in the aerospace, rail and defence sectors identify design and safety processes for different Safety Integrity Levels (SILs) or Development Assurance Levels (DALs). The assumption is that components developed against the requirements of higher SILs/DALs will be less prone to critical failures and thus have a lower impact on the safety of the overall system. This paper questions this assumption and instead discusses assurance of the safety argument as a method of demonstrating the confidence that can be placed in a safety case. An industrial case study from the aerospace sector is presented to demonstrate the practical use of the concept.

Keywords: Safety Arguments, Assurance

1 Introduction

Many standards across the Defence, Aerospace, Transportation and Nuclear sectors require the development of a safety case for safety critical systems. Such safety cases are typically developed by the design authority and accepted by another (regulatory) authority. Implicit in this process is the assessment of whether the safety argument within the safety case has been *sufficiently* assured with the evidence available. However, the implicit determination of the confidence in a safety case can lead to the degree of subjectivity in the development and acceptance being greater than desirable. This paper presents a new approach for considering and explicitly describing safety case assurance. The approach described is a process which occurs in the development and assessment of a safety case but currently remains unexpressed.

Copyright © 2003, Australian Computer Society, Inc. This paper appeared at the *8th Australian Workshop on Safety Critical Systems and Software (SCS'03)*, Canberra. Conferences in Research and Practice in Information Technology, Vol. 33. P. Lindsay & T. Cant, Eds. Reproduction for academic, not-for profit purposes permitted provided this text is included.

1.1 Current Standards

Currently, standards determine the confidence that can be placed in the safety of a system by assuring the development process. Most standards used to guide the development of software for safety critical systems, e.g. DO178B (RTCA/EUROCAE 1992), DS 00-55 (UK MoD 1996), and Part 3 of IEC 61508 (IEC 1999), identify processes for different safety integrity levels (SILs) or development assurance levels (DALs). Both the developer and assessor accept that, by following the process of applying these techniques and developing evidence, the software achieves the required level of safety. However, the safety evidence generated does not necessarily give a quantitative demonstration that the SIL or DAL has been achieved. Also, due to the difficulty in detecting software failures in accidents, the commercial sensitivity of failure data, and the extremely high safety levels required of software, it is difficult to determine the operational levels of safety for developed software. Thus it is often not possible to assess before or during operation whether software produced to a SIL or DAL process attains the required level of safety. There is some evidence to show that software developed by a process-based approach may not always meet the required level of assurance (Harrison 1999). The extent to which there is evidence that the approaches advocated by these standards are effective in practice has previously been questioned (Lindsay & McDermid 1997, McDermid & Pumfrey 2001).

1.2 Assurance of the Safety Argument

The authors of this paper advocate the development of a safety argument to determine the rationale behind the selection of safety evidence (Kelly 1998, Weaver & McDermid 2002). The aim of the safety argument within a safety case is to clearly demonstrate how the evidence meets the safety requirements. The Goal Structuring Notation (GSN), (Kelly 1998), is a graphical notation for constructing complex safety arguments for safety cases. GSN has already been widely adopted on a number of large scale projects across the Defence, Aerospace, Transport and Nuclear industries.

Instead of attempting to assure the development process, this paper considers the assurance of a GSN safety argument within the safety case as an approach for deriving the confidence that can be placed in the safety

case and the evidence presented. Using GSN, it is possible to build both strong and weak safety arguments, and this strength is based upon the extent to which the safety requirements have been satisfied by the evidence. Safety Assurance Levels (SALs), described in this paper, articulate the judgements made by argument developers about the relevance of individual items of evidence and the completeness of the evidence set. A comparison is made between SALs and existing industrial approaches.

2 Argumentation in Safety Cases

Argumentation
The action or operation of inferring a conclusion from propositions premised
Oxford English Dictionary (Oxford 1991)

An argument, in its most basic form, is an inference from one or more premises (also known as propositions or grounds) to a conclusion (also known as claims).

Premise
A previous statement or proposition from which another is inferred or follows as a conclusion
Oxford English Dictionary (Oxford 1991)

Conclusion
A judgement or statement arrived at by any reasoning process
Oxford English Dictionary (Oxford 1991)

GSN argument structures, which also include context, strategy, justifications and assumptions, are built by linking together premise(s)-conclusion inferences (figure 1), where the conclusion of one argument step becomes the premise of the next.

System safety requirements are complete
System safety requirements are met

System is acceptably safe

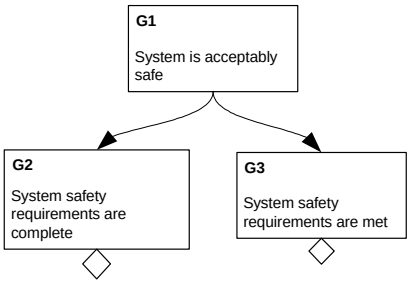

Fig. 1. Argument representations in Premise/Conclusion format and GSN

In GSN, premises and conclusions are represented as goals. Premises can be supported by evidence, which is represented as solutions in GSN, and/or decomposed into further premises (goals). Using this approach, large and detailed arguments can be created for a single top-level goal (or claim). Items of safety evidence support basic premises, while top-level goals usually relate to the safety requirements of the system. Figures 2 describes the principle elements of GSN and Figure 3 gives an example goal structure

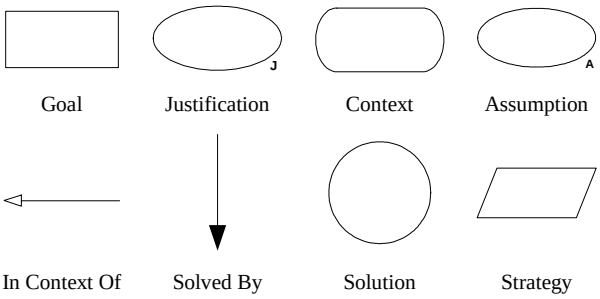

Fig. 2. Principle Elements of GSN

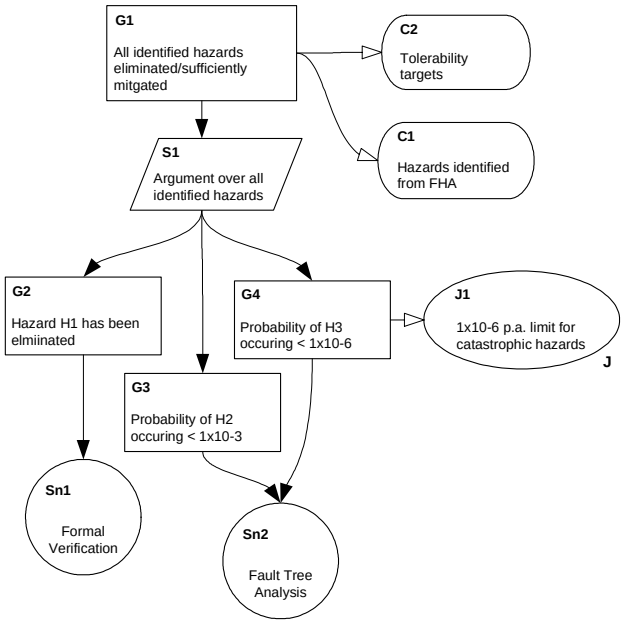

Fig. 3. Example Goal Structure

The statements made in conclusions and premises (and hence GSN goals) are propositions. These propositions can be qualitative or quantitative and may be subjective in nature. However, the statements are either true or false. For example, the statement "50% of the population is female" is either a true or false statement, as is the statement "failure rate of component X is 10^{-4} failures per operational hour". This characteristic of statements leads to arguments having properties based upon the truth or falsity of the statements. In argumentation, the strongest arguments are designed to be both Valid and Sound.

Valid
If premises are true, conclusion is true
Philosophical Logic – An Introduction (Wolfram 1990)

Sound
Argument which is valid and has true premises
Philosophical Logic – An Introduction (Wolfram 1990)

It is desirable to develop safety arguments that are both valid and sound. However, due to the evidence typically

available and the inferences that must be made, a provably valid and sound argument is unobtainable for a Safety Critical System. Thus, the goal structure notation accepts arguments that are consistent and thus causally weaker.

Consistent
If premises are true, conclusion may be true
Philosophical Logic – An Introduction (Wolfram 1990)

This weaker form of causal relationship is known as *inductive* argumentation, while the stronger, valid argument form is known as *deductive* argumentation.

Deductive Argument
If premises are true, then the conclusion must also be true
The Philosopher's Toolkit (Baggini & Fosl 2003)

Inductive Argument
The conclusion follows from the premises not with necessity but only with probability
The Philosopher's Toolkit (Baggini & Fosl 2003)

The inductive nature of GSN implies that a level of probability must be associated with the satisfaction of a safety argument. It is not the case for goal structures that the top-level goal is true because all of the solutions are true. Instead, the aim of the argument is to show the *sufficiency* of the child goals and solutions in satisfying the parent goal. While GSN describes the relationship between premises and conclusions, it does not capture the inductive nature of the safety argument. In GSN strategy elements can be used to annotate the relationship between parent goals and child elements. This aids the reader in judging the strength of the premises-conclusion relationship. However, the level of support remains implicit.

For inductive arguments it can be useful to express the relevance of each child element in satisfying the parent goal and the strength of the argument step as a whole. It is beneficial to make explicit the connectivity within the causal relationships between parent goals and child goals/solutions. This will clarify the sufficiency of the premises (solutions) in satisfying the conclusion (top-level goal). By making explicit the *strength* of the argument the knowledge captured within the goal structure will be increased. Thus the argument is both improved and made more transparent.

3 Assurance of Arguments

Assurance
Subjective certainty, a being certain as to a fact, certitude; confidence, trust
Oxford English Dictionary (Oxford 1991)

The term "Assurance" is used within this paper for discussing the strength of a safety argument. Assurance inherently expresses the subjectivity when determining the strength of an inference. It also encapsulates the concept of confidence, which is part of the objective of a safety argument – the determination of the confidence that can be placed in the safety of a system. Assurance is a property of an argument's conclusion. It is based upon:

- the likelihood that the premises are true (i.e. the assurance of the premises); and
- the extent to which the premises entail[1] the conclusion

The overall assurance of a safety argument is equal to the assurance of the top-level goal of that argument.

Safety Assurance
A qualitative statement expressing the degree of confidence that a safety claim is true.
Working Definition

The size and complexity of safety arguments combined with the subjective nature of argument composition is such that assurance cannot easily be considered quantitatively. For example, a Bayesian approach (Fenton et al 1998) makes the heavy demand that the relationship between all premises and conclusions can be expressed as a conditional probability. Instead, we believe a qualitative approach, expressing levels of assurance, similarly enables articulation of the strength of arguments without creating an unreasonable burden on argument creator or assessor. Assessment of assurance can be a qualitative judgement based upon an understanding of the child element to parent goal inference. By expressing the assurance, these judgements are made explicit within the argument, allowing other readers to agree or disagree.

To provide a framework for communicating and assessing these judgements, levels of assurance will be used. The primary reason for the discussion of assurance in terms of levels is to act as a coarse quantification. This aids judgement when assessing an argument as a whole and allows tolerance between slight variations in opinion. From a safety argument developer's point of view, the use of levels clarifies where the focus of effort is required for evidence generation. From the certifier's point of view, levels clarify the important aspects of the argument and from a management perspective, levels give a shorthand for understanding time and financial costs of the argument creation. There is no particular number of levels that should be used.

4 The Process of Applying Safety Assurance Levels (SALs)

Safety case construction should start at the beginning of the safety lifecycle and continues until completion of the safety lifecycle and the production of the final safety case document (Kelly et al 1997). Within the lifecycle, the common key stages of safety case construction are the creation of a preliminary safety case and the production of the final safety case. Preliminary safety case

[1] To involve, logically necessitate (a particular conclusion) (Oxford 1991)

construction begins with the determination of the system safety requirements based upon hazard identification and risk assessment. The preliminary safety argument is formed from a top-down decomposition of these safety requirements. The final safety case requires a bottom up confirmation of the safety argument that the safety evidence generated meets the safety requirements identified.

Safety Assurance Level (SAL) apportionment runs in step with the evolution of the safety argument. During preliminary safety case construction a speculative determination of the SAL for the argument as a whole and then for each goal occurs in a top-down fashion. This process of safety argument creation and SAL apportionment together determines both the nature and strength of the evidence required. During final safety case production confirmation is provided that the safety evidence meets the safety requirements via a bottom up reading of the argument and the assurance of the argument steps. The apportionment of SALs demonstrates the sufficiency of the argument and provides confidence that the safety requirements have been met. There are three stages to SAL apportionment:

- Setting Top-level Safety Assurance Levels
- Parent Goal-Child Goal(s)/Solution(s) SAL decomposition
- Determining SALs for Evidence

These three stages will be discussed in the following sections.

4.1 Top Level SAL Determination

An argument is created in an attempt to prove the truth of a statement. This statement is the objective and top-level goal of the argument. Determination of the SAL for the top-level goal sets the required or target assurance of the argument. Safety arguments can be based upon many different parts and aspects of Safety Critical Systems (e.g. the system, a component, a hazard). Establishing the required assurance for an argument is based upon the severity associated with the failure of the top-level objective. For probabilistic arguments when determining the top-level SAL, the acceptable level of risk must be established. Many standards give guidance on assessing risk, for example (RTCA/EUROCAE 1992, UK MoD 1996, IEC 1999, CENELEC 1999).

4.2 Argument Decomposition and the Support Type

When developing an inductive argument, there is a requirement for the argument to be cogent.

> *Cogent*
> The premises give good rational support to the conclusion
>
> A Practical Study of Argument (Govier 1988)

Within GSN, the SAL assigned to the parent goal determines the required cogency of the argument step from child to parent goal. Using GSN, arguments are typically constructed in a top-down fashion such that suitable premises are developed which show the acceptability of the conclusion. Child goals and/or solutions that support the goal are identified and these child elements must be suitable to produce a cogent argument. The level of support is determined by the assurance of the child elements and the extent to which the child elements entail the conclusion or Parent Goal. In SAL argument decomposition, the minimum required assurance level of each child element is determined. The SAL decomposition process determines the level of assurance that is required of the child elements in order to sufficiently assure the parent goal. Once the goal structure is completed confirmation can be provided (bottom up) that the evidence referenced sufficiently assures the basic premises of the argument and that those premises ultimately sufficiently assure the conclusion of the argument. In this way confirmation can be provided that the argument is cogent and the top-level goal has been sufficiently satisfied.

The first stage to determining argument sufficiency is to establish the relevance of each individual child element to the parent goal.

> *Relevance of Child Elements*
> The extent to which the child element entails the parent goal
>
> Working Definition

A child element on its own can either totally satisfy the entire parent goal or can partially satisfy the parent goal. Argument support provided by the child element set can have one of three forms. Govier, in (Govier 1988) identifies these three types of argument support:

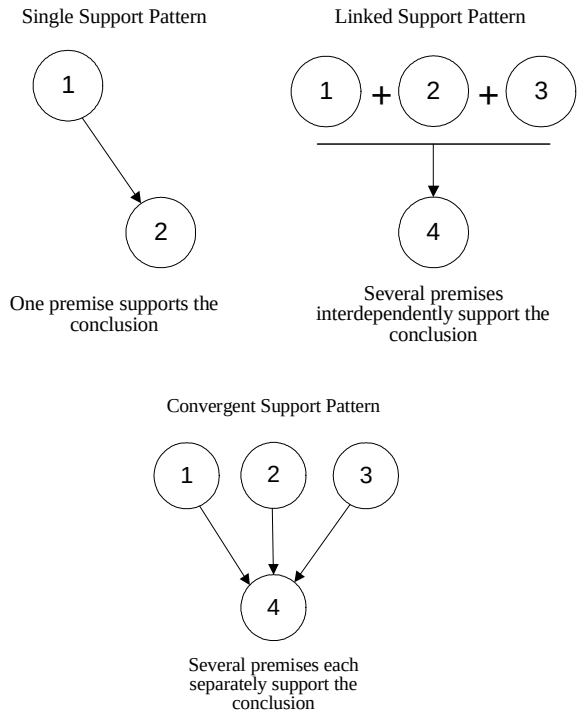

Fig. 4. Govier's Support Pattern Types (Govier 1988)

A child element that satisfies the entire parent goal provides single support. The relevance of a child element is determined, by considering whether the truth of that child element statement entails the truth of the parent goal statement.

RELEVANCE OF CHILD ELEMENT

If child element {GX} was true and all other child elements removed, could the parent goal still be satisfied?

YES – The child element {GX} fits a *single support* pattern

NO – Other child elements are required as well as {GX}

Highly relevant child elements which each fit a single support pattern can be used in combination to produce a convergent support pattern. A child element that does not fit a single support pattern may, in combination with other child element(s), fit a linked support pattern. With a convergent support pattern each child element independently addresses the whole of the parent goal. With a linked support pattern each child element addresses a different aspect of the parent goal. Figure 5 shows a good example of a linked support pattern. In the example each of the child goals (G2, G3, G4) addresses a different aspect of the parent goal. Each child goal is required and the argument could not be supported if any of the child elements were removed. For example, if G2 was removed it would be impossible for the parent goal G1 to be satisfied.

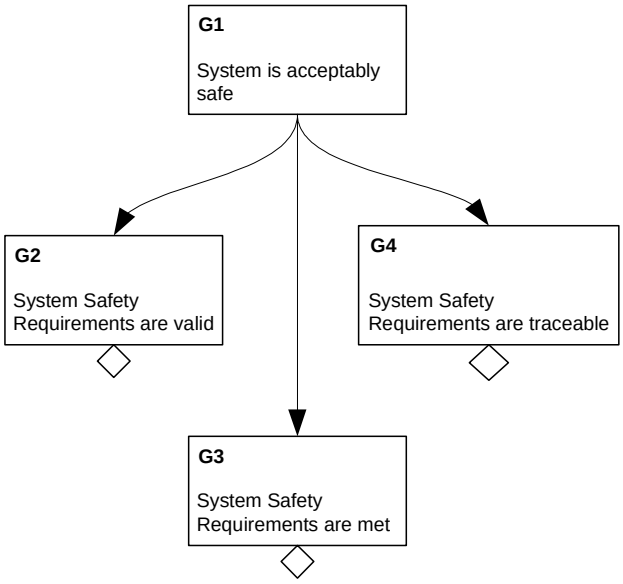

Fig. 5. A Linked Support Argument

Figure 6 shows a good example of a convergent support pattern. In the example each of the child goals (G2, G3) addresses the whole of the parent goal. Each child goal is capable of satisfying the parent goal separately. For example G2 is capable of satisfying the parent goal G1 alone, as is G3.

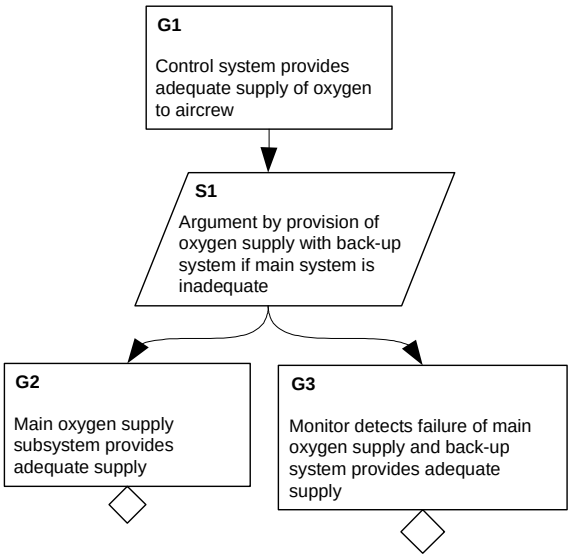

Fig. 6. A Convergent Support Argument

Child elements that fit a convergent pattern are identified by assessing the relevance of the child element. Child elements that fit a linked support pattern are identified by determining whether the element is required. Child elements that (as a set) do not fit a linked or convergent support pattern can be described as fitting a hybrid support pattern, which is neither linked nor convergent.

REQUIREMENT FOR CHILD ELEMENT

If child element {GX} was removed and all other child element's were true, could the parent goal still be satisfied?

YES – The child element {GX} is one of the following:

{GX} forms part of a *convergent support* pattern
{GX} forms part of a *hybrid support* pattern

NO – The child element is *required* and fits a linked support pattern

4.3 Re-factoring the Argument to fit a Support Pattern

Within an argument step it is beneficial that all child elements fit a linked support pattern, or all child elements fit a convergent support pattern. This aids the production of a clear and understandable argument and helps to determine the required assurance of the child elements. A structure that is neither linked nor convergent can appear ambiguous in how it satisfies the parent goal. It can be difficult to discriminate between distinct threads of argument and those that reinforce one another. There are two possible forms of hybrid support that can be removed by restructuring the argument. Arguments which contain delineated convergent and linked support can be broken down into smaller stages, as shown in figures 7 and 8. These restructured arguments are convergent with a linked sub-argument clearly identified.

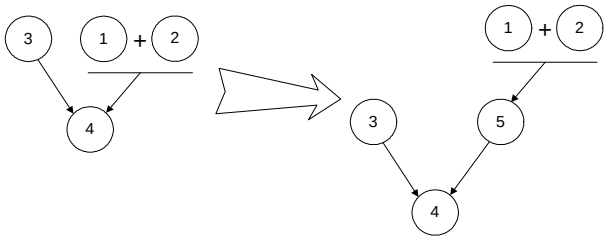

Fig. 7. Argument broken down into smaller stages using Govier's Notation

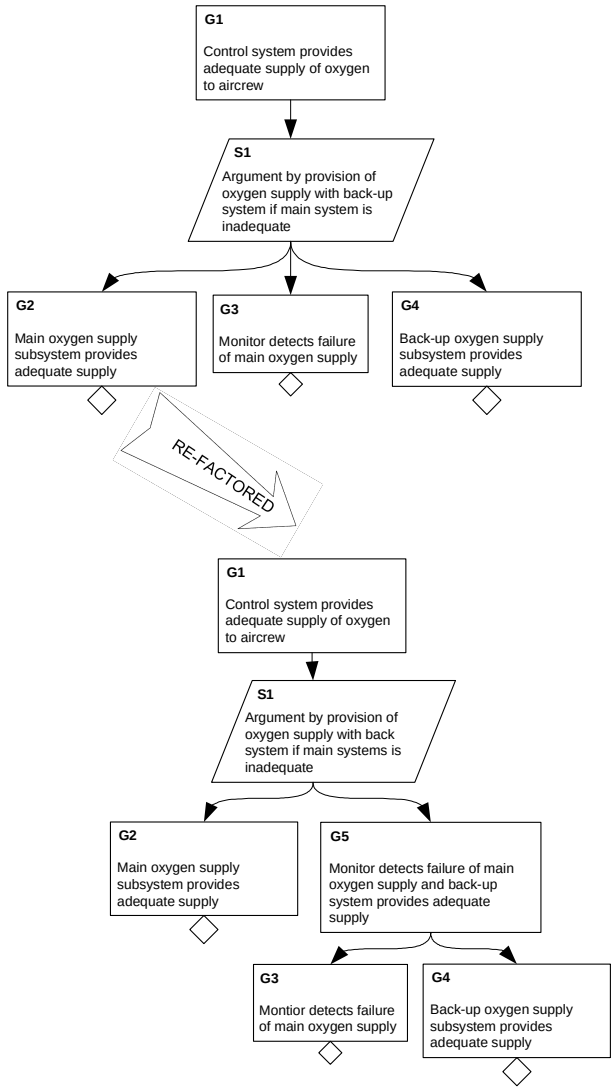

Fig. 8. Argument broken into smaller stages using GSN

Arguments that contain overlapping linked support must be broken down by including an intermediary stage within the argument, as shown in figures 9 and 10. In figure 10 there are aspects of the test sets 1 and 2 that are distinct and aspects that overlap – however this is unclear from the initial argument. The restructured arguments are linked with a convergent sub-argument clearly identified.

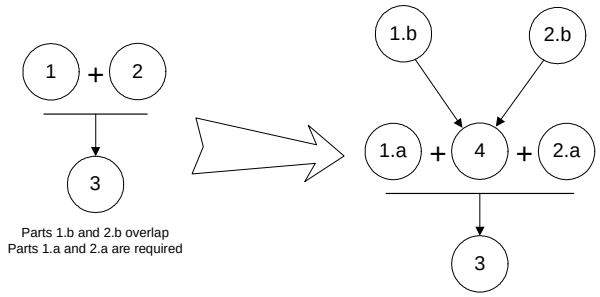

Fig. 9. Intermediary step placed in argument using Govier's Notation

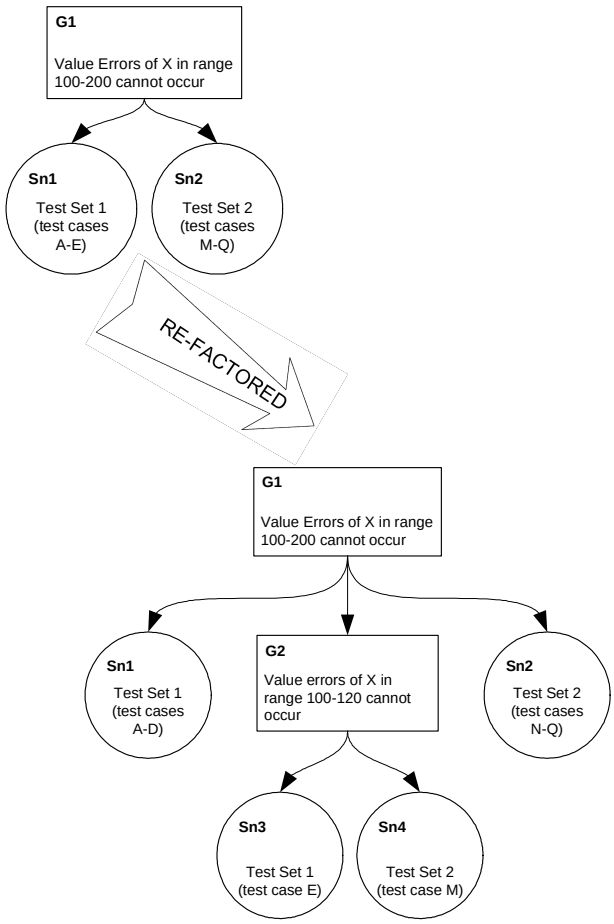

Fig. 10. Intermediary step placed in argument using GSN

Before considering SAL decomposition across child elements, it is important to identify the support form of the argument step. Any necessary refactoring of the argument must be performed such that any argument step provides either linked or convergent support.

4.3.1 Trade off between claims and assurance

When applying the SAL approach to existing safety arguments the tradeoff that can be made between the nature of the claim and the assurance required becomes apparent. For example, it can often be the case that a bold and broad claim (such as "failure mode X will never occur") can only be weakly assured. Whereas a narrower claim (such as "failure mode X will not occur in the next 3 years") can be more strongly assured. This highlights the tradeoff between the nature of the claim put forward

and the ease with which the required level of assurance can be demonstrated. A positive side-benefit of reasoning about SALs is that it promotes precision in the statement of safety case objectives.

4.4 SAL Decomposition across Child Elements

SAL decomposition depends upon the type of support provided by the child elements (linked or convergent), the assurance required by the parent goal (i.e. the SAL of the parent goal) and the relevance of the support to the conclusion. For the description of SAL decomposition in this paper, four levels of assurance are used. This is to maintain the style used by both Safety Integrity Levels (SILs) in (UK MoD 1996, IEC 1999 and CENELEC 1999) and Development Assurance Levels (DALs) in (RTCA/EUROCAE 1992). As with SILs the highest level is 4 with the lowest level being 1. Where there is no diversity in the argument support, assurance is directly proportional to relevance and thus four levels of relevance are used: Valid/Near Valid, High, Medium and Low.

4.4.1 Linked Support

If the argument step fits a linked support pattern, each child elements supports part of the parent goal which no other child goal or solution supports. If the child element set has a valid/near valid relevance to the parent goal, the coverage of the parent goal is total or near total. This means that each child element must maintain the SAL of the parent goal. By maintaining the SAL of the parent goal, coverage is maintained down the steps of argument. If SALs were reduced in this situation, each step in the argument would allow a lower coverage within the child elements, thus reducing the coverage of the solutions with respect to the top-level goal.

For parent goals which are SAL 4, the relevance of the linked support child elements to the parent goal must be valid or near valid. They must maintain the level of assurance and thus all child elements must be SAL 4. For parent goals that are SAL 3, 2 or 1, if they provide total coverage of the parent goal, they must also maintain the SAL of the parent goal. However, for goals which are SAL 3, 2 or 1, full assurance is not necessarily required, thus it is acceptable to have reduced relevance with respect to the parent goal.

For SAL 3 Parent Goals, if the relevance is reduced to high, the child elements must be assured to SAL 4 (table 1). The requirement to have SAL 4 child elements prevents loss of coverage within the argument steps below the step being considered.

For SAL 2 Parent Goals, if the relevance is reduced to medium, the child elements must be assured to SAL 4 (table 1). As with SAL 3 parent goals, the requirement to have SAL 4 child elements prevents loss of coverage within the argument steps below the step being considered.

For SAL 1 Parent Goals, if the relevance is reduced to low, the child elements must be assured to SAL 4 (table 1). As with SAL 3 and SAL 2 parent goals, the requirement to have SAL 4 child elements prevents loss of coverage within the argument steps below the step being considered.

If the relevance of the child elements is reduced, the concession for a SAL 3, 2 or 1 argument is made at this stage in the argument decomposition, and thus support for these child elements must maintain a valid relevance (SAL 4). As relevance is directly proportional to assurance in linked support arguments this means that for SAL 2 parent goals, high relevance child elements require SAL 3 child elements (table 1). Similarly SAL 1 parent goals require SAL 3 child elements when they are of medium relevance and SAL 2 when they are of high relevance (table 1).

Parent SAL	Relevance	Child SAL
S4	Valid/Near Valid	S4
S3	Valid/Near Valid	S3
	High	S4
S2	Valid/Near Valid	S2
	High	S3
	Medium	S4
S1	Valid/Near Valid	S1
	High	S2
	Medium	S3
	Low	S4

Table 1. Determining Child Element SALs for Linked Support

This table is only a reflection of a possible SAL decomposition approach. They can only be used in combination with further justification, as discussed in section 4.4.3

4.4.2 Convergent Support

With convergent support, the assurance of the parent goal is split across the child elements. Each child element supports the parent goal independently and thus it is important for this support type to identify the independence of the child elements.

Independence of Child Elements
The extent to which complementary child elements follow diverse approaches in fulfilling the parent goal
Working Definition

Independence can be Conceptual or Mechanistic. Conceptually different approaches are based on different underlying theories. For example static and dynamic analysis are conceptually different approaches to developing evidence (one involves running the program; the other does not). Mechanistically different approaches implement the same underlying theory in different ways. For example the same testing technique performed by two

different testing teams is mechanistically different. As a general rule conceptual independence is more significant than mechanistic independence. For goals that require higher assurance, conceptual independence of the child elements is required, whereas for lower SAL goals mechanistic independence is acceptable.

The assurance of convergent support child elements is dependent upon the focus of the argument. The argument can be constructed to rely equally upon all child elements or can be focused to rely on certain child elements more heavily (table 2). For joint responsibility the child elements are all required to have the same SAL. For convergent support with an argument focus, the SAL of certain child elements is greater than of others as they provide the main focus of the argument.

Parent SAL	Complementarity		Child SAL	Child SAL
	Independence	Child elements		
S4	Conceptual	2	S2	S4
	Conceptual	2	S3	S3
S3	Conceptual	2	S1	S3
	Conceptual	2	S2	S2
S2	Mechanistic	2	S1	S1

Table 2. Determining Child Element SALs for Convergent Support

This table is only a reflection of a possible SAL decomposition approach. They can only be used in combination with further justification, as discussed in section 4.4.3

The decision for joint responsibility or argument focus is usually based upon design and other dependability considerations as well as an understanding of what evidence can be generated for the different items of support. The tables show how the SAL decomposition can be performed for two convergent child elements, but can be expanded for support which contains further child elements.

4.4.3 Tables and Meta-arguments

The tables in this paper act as guidance and do not provide an exact rule for SAL decomposition. For each argument step a justification must be given which expresses the rationale behind the SAL decomposition. Within the justification, it is also necessary for the developer to make explicit what the levels of relevance equate to. The "justification" element type within GSN "provides the rationale behind the adoption of some strategy or the presentation of some goal" (Kelly 1998). They are simple prepositional backing statements concerning the argument. To make justifications about the validity of arguments and the determination of goal/solution SALs, we require justifications that are more complex than the GSN justification element allows. These more complex justifications can be developed in the form of meta-arguments.

Meta-arguments can provide a secondary justification to any aspect of a safety argument and are represented in the GSN format. They can be used to provide additional contextual information in addition to the central 'spine' of the safety argument. By using meta-arguments, the amount of information that can be captured in the complete safety argument is increased without increasing the complexity of the main argument's structure. Thus, additional justification of the argument can be included, whilst retaining the clarity of the primary safety argument.

4.5 Determining Evidence Safety Assurance Levels

The final stage in SAL apportionment is determination of the SAL of an item of evidence. After determining the Top-level SAL and decomposing this throughout the safety argument structure, the solutions will have a required SAL which must be shown to be met. The SAL of a solution is an expression of the process evidence related to generating the evidence. The evidence's SAL identifies the trustworthiness of the evidence and is thus based upon a number of factors. These factors include, but are not limited to:

- "Buggy-ness" – how many "faults" there are in the evidence presented
- Level of review
- For hand-generated evidence: Experience and Competency of Personnel
- For tool-derived evidence: Tool Qualification and Assurance
- Competence of the personnel

5 Industrial Case Study

This example industrial case study considers a simplified system typically encountered in many domains. It has been derived from a real system, in the aerospace sector, and the details abstracted. The system is one in which an operator monitors a display system to decide when to sanction a safety-critical action via an input to the control system (figure 11). As part of this system, human reasoning is required to determine, from the displayed information, and the system environment, when it is safe to perform the action. Activation of the system in an incorrect environment could lead to a catastrophic hazard.

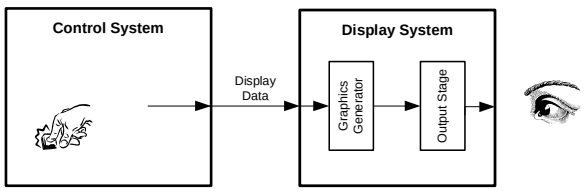

Fig.11. Basic System, comprising Control, Display and Operator

The hazard severity would suggest, assuming no external mitigation is possible or at least sufficient to reduce the

requirement, that the claim about the system as a whole would inherit a Safety Assurance Level of 4. The simplest solution would be to allocate this SAL to all claims about components, in respect of demonstrating adequate mitigation of this hazard (figure 12). This argument decomposition, through strategy S1, follows a linked support pattern whereby each child goal supports an independent part of the parent goal. Similarly the strategies S2 and S3 also provide a linked support decomposition. Thus each child goal maintains the SAL of the parent goal (SAL 4) and the set of child goals must have complete, or valid, relevance with respect to the parent goal. It is reasonable to expect that the claim about the control system should inherit this SAL 4 as it will perform the safety-critical action. It should be noted that the operator claim also inherits a SAL 4 which will need to be justified in the safety argument through procedures and training, etc. However, the feature of interest in this paper is the allocation of a SAL to the claim about the display interface regarding the correctness and validity of the data displayed on it (G3). Unless the display is dedicated and bespoke, the complexity of modern display systems mean that it is unlikely that sufficient confidence could be gained or evidence generated to support a SAL 4 claim for the display.

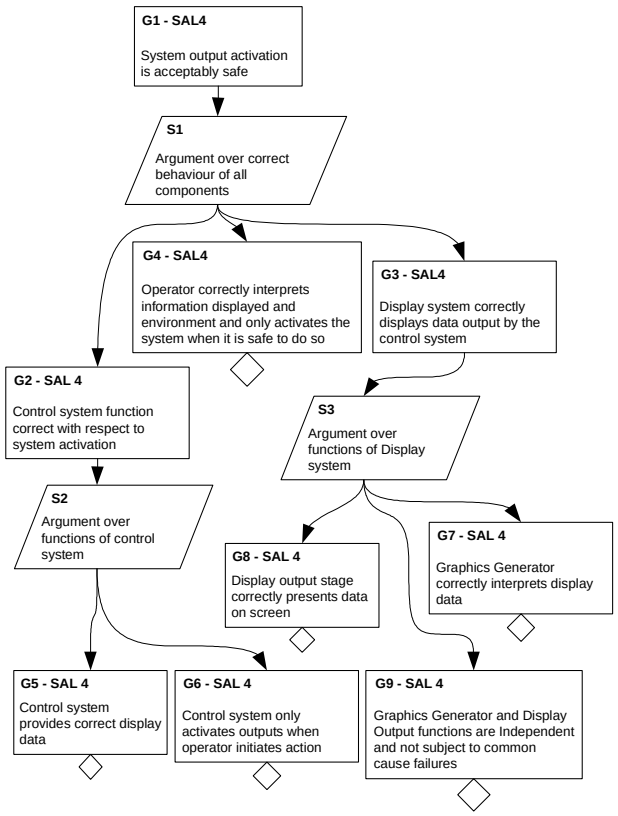

Fig.12. Safety Argument for Basic System

In this example, the display is a Commercial Off The Shelf (COTS) component, and consequently, alternate strategies will need to be employed. One solution might be to provide a comparison function to check that what has been displayed to the operator is that which the control system had transmitted for display. This could be supported by the control system which would provide sufficiently independent checking functions (figure 13).

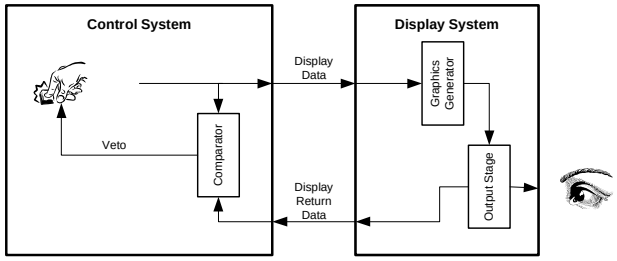

Fig.13. Practical System Solution including Comparator

The system is then reliant on the control system providing the comparison of display and display return data and on the independence of the software display graphics generator function and the hardware output stage. However, as there is no check on the actual output on the display surface, the assurance required of the hardware in the display system in this implementation remains at the higher level of assurance (figure 14).

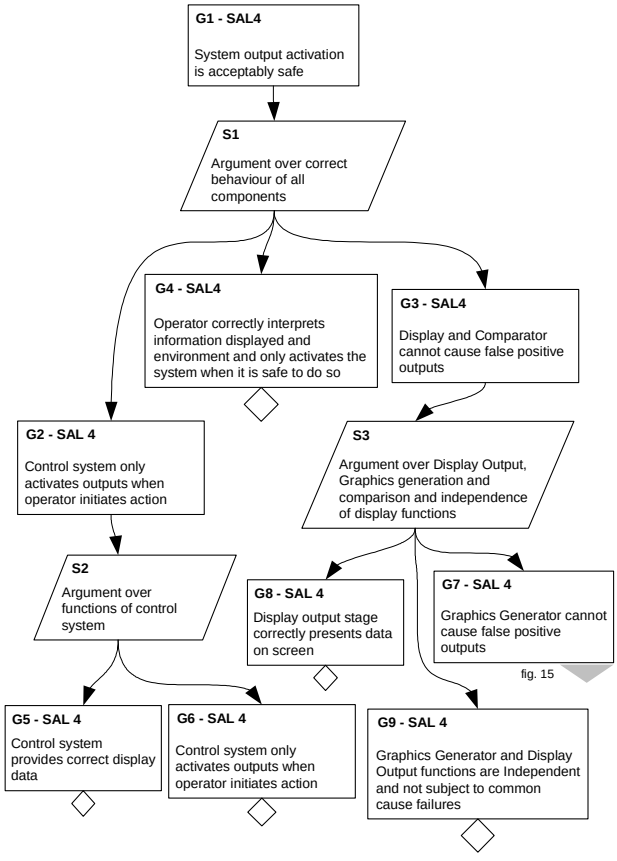

Fig.14. Top level of Argument for Practical Solution

Again in this decomposition the strategies S1, S2 and S3 represent linked support decompositions of the goals G1, G2 and G3 respectively. Each child goal maintains the SAL of the parent goal. Goal G7 is decomposed further in figure 15. Suitable evidence to support the independence argument might include a Failure Modes and Effects Analysis, (FMEA), for example, which shows that no single failure could cause complementary corruptions which might lead to the generation of a 'false positive' comparison of display and display return data at

the comparator, even though the display did not show what was requested by the control system.

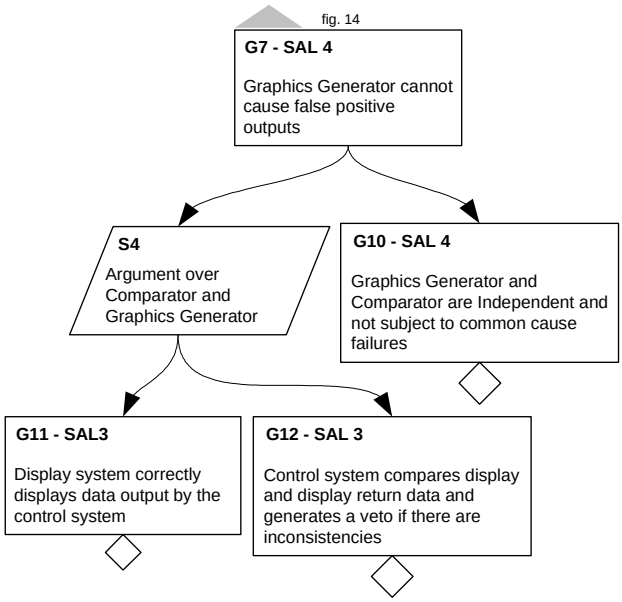

Fig.15. Decomposition of Goal G7

In figure 15, the Goal G7 is supported by a convergent argument decomposition (S4), whereby the goal is supported by two valid claims each of which can, on its own, address the parent goal. The decomposition requires an argument that the sub goals are independent and this is included (G10) at the level of SAL 4.

This argument requires only the claims about the Hardware Output Stage of the Display system to be at the higher level of assurance, while the Software Graphics Generator claim is assured at the lower level. With this argument approach neither the comparator nor the Graphics generator provide a single point of failure, and their combined use provides a high level of assurance (SAL 4).

6 Comparison with Other Concepts

SALs describe the level of confidence that can be placed in an argument. Safety Integrity Levels (SILs) (UK MoD 1996, IEC 1999 and CENELEC 1999) and Development Assurance Levels (DALs) (RTCA/EUROCAE 1992) imply a level of system dependability from assurance of development and assessment processes. As SILs and DALs are assigned to systems, components or functions they consequently dictate processes at a high level. This leads to the use of SILs and DALs being criticized as they necessitate a process which requires a broad set of evidence to be generated without consideration of the specific objectives required for a compelling safety case. Similarly, Assurance Evidence Levels (AELs) (CAA 1999) determine the nature of evidence required at a *component* level. Therefore this approach does not necessarily target the evidence selection towards specific safety case objectives. While Def(Aust) 5679 (Australian DoD 1998) does assign SILs to Component Safety Requirements, it is still only at the component level and thus does not allow the assurance to be decomposed further than this. Safety Assurance Levels are the only concept which assigns assurance to safety case claims.

SALs, by being associated with particular claims of the safety argument, focus consideration of assurance and selection of individual items of evidence specifically against the primary issues of concern for a certain system application. By enabling reasoning of how assurance emerges within argument structures, SALs also can be used prior to evidence selection to aid focusing and transparent structuring of the safety case argument.

SALs can be attached to all types of evidence. For example independence evidence for SIL 4 decomposition across two SIL 3 components and a SIL 4 combinator in DS 00-56 (UK MoD 1996) demands independence evidence, however it is not possible using SILs to associate a required integrity with that independence evidence. SALs allow a required level of assurance to be attached to the independence evidence as well as the component evidence.

Within a safety argument, decomposition is not complete until a requirement for a specific item of evidence is determined. By using SALs the required assurance of the evidence is set at this stage and thus each item of evidence will have its own required level of thoroughness. SALs justify the concentration of effort upon parts of the argument that require greater assurance. This provides a more rational approach to the selection of evidence based upon the argument being generated and allows safety engineers to produce suitable evidence of the correct weight.

Bayesian Belief Networks (BBNs) (Fenton et al 1998) offer an approach for describing the causal relationships between entities or nodes. The strength of the relationship between node values is embodied as conditional probabilities within Node Probability Tables (NPTs). However, the strength of the relationship is not easily communicated and the values contained within the NPTs can be initially difficult to elicit (e.g. from experts) and to validate. SALs, when used with GSN, explicitly present how the safety case author believes assurance is established within a safety argument. Unlike BBNs, there is no automated calculation of assurance, based upon the strength of inferences. Instead, the author is encouraged to explicitly reason about assurance decomposition using concepts found in argumentation theory (support type, relevance, independence) and where necessary justify decomposition using meta-arguments.

7 Conclusions

This paper describes a principle which underlies the development and acceptance of all safety cases. The determination of the level of assurance which a safety case provides is an evaluation which is currently being made implicitly. The role of safety case assurance is too important for it to remain implicit within a large and complex safety argument. This paper has described an approach for determining and explicitly presenting this confidence. By including assurance within a GSN safety argument, the role each item of evidence plays is expressed more clearly. It is possible to determine what

items of evidence satisfy an objective and also how valuable each item is in satisfying that objective. By using SALs during argument construction, the required coverage and quality of evidence is expressed more thoroughly. This aids evidence selection, ensuring that the correct balance within the evidence set can be achieved. SALs allow the inclusion of process-based information about the evidence to be captured, which underpins the product-based argument. When reviewing arguments that contain SALs, the relationship of each item of evidence to the objectives will be clearer, and thus the focus of the argument will be more comprehensible.

8 References

Australian DoD (1998): *The Procurement of Computer-Based Safety-Critical Systems Def(Aust) 5679*. Australian Department of Defence.

Baggini, J., Fosl, P.S. (2003): *The Philosophers Toolkit – A Compendium of Philosophical Concepts and Methods*. Blackwell.

CAA (1999): *Regulatory Objective for Software Safety Assurance in Air Traffic Service Equipment SW01*. Civil Aviation Authority.

CENELEC (1999): *Railway applications – Safety related electronic systems for signalling*. European Committee for Electrotechnical Standardisation.

Fenton, N., Littlewood, B., Neil, M., Strigini, L., Sutcliffe, A., Wright, D. (1998): Assessing Dependability of Safety Critical Systems using Diverse Evidence. *IEE Proc. Software Engineering*, **145**(1): 35-39.

Govier, T. (1988): *A Practical Study of Argument.* Wadsworth.

Harrison, K.J. (1999): Static Code Analysis on the C-130J Hercules Safety Critical Software. *Proc. 17th International System Safety Conference*, Florida, USA, System Safety Society.

IEC (1999): *IEC-61508: Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems*. International Electrotechnical Commission.

Kelly, T. (1998): *Arguing Safety – A Systematic Approach to Safety Case Management*. DPhil Thesis, Department of Computer Science, University of York.

Kelly, T.P., Bate I.J., McDermid, J.A., Burns, A. (1997): Building a Preliminary Safety Case: An Example from Aerospace, *Proc. 1997 Australian Workshop on Industrial Experience with Safety Critical Systems and Software*, Australian Computer Society, Sydney, Australia.

Lindsay, P.A., McDermid, J.A. (1997): A Systematic Approach to Software Safety Integrity Levels. *Proc. 16th International Conference on Computer Safety* (SAFECOMP 97),York, UK.

McDermid, J.A., Pumfrey, D.J. (2001): Software Safety: Why is there no Consensus? *Proc. 19th International System Safety Conference*, Huntsville USA, System Safety Society.

Oxford (1991): *Oxford English Dictionary*. Oxford University Press.

RTCA and EUROCAE (1992): *Software Considerations in Airborne Systems and Equipment Certification*. Radio Technical Commission for Aeronautics, RTCA DO-178B/EUROCAE ED-12B.

UK MoD (1996): 00-55 *Requirements of Safety Related Software in Defence Equipment*. Ministry of Defence, Defence Standard

UK MoD (1996): *00-56 Safety Management Requirements for Defence Systems*. Ministry of Defence, Defence Standard.

Weaver, R.A., McDermid, J.A. (2002): Software Safety Arguments: Towards a Systematic Categorisation of Evidence. *Proc. 20th International System Safety Conference*, Denver USA, System Safety Society.

Wolfram, S. (1990): *Philosophical Logic – An Introduction*. Routledge.