The Goal Structuring Notation – A Safety Argument Notation

Tim Kelly and Rob Weaver
Department of Computer Science and Department of Management Studies
University of York, York, YO10 5DD UK
tim.kelly@cs.york.ac.uk, rw24@york.ac.uk

Abstract

In Europe, over recent years, the responsibility for ensuring system safety has shifted onto the developers and operators to construct and present well reasoned arguments that their systems achieve acceptable levels of safety. These arguments (together with supporting evidence) are typically referred to as a "safety case". This paper describes the role and purpose of a safety case. Safety arguments within safety cases are often poorly communicated. This paper presents a technique called GSN (Goal Structuring Notation) that is increasingly being used in safety-critical industries to improve the structure, rigor, and clarity of safety arguments. The paper also describes a number of extensions, based upon GSN, which can be used to assist the maintenance, construction, reuse and assessment of safety cases. The aim of this paper is to describe the current industrial use and research into GSN such that its applicability to other types of Assurance Case, in addition to safety cases, can also be considered.

1. Introduction

The purpose of a safety case can be defined in the following terms: *A safety case should communicate a clear, comprehensive and defensible argument that a system is acceptably safe to operate in a particular context.*

The concept of the 'safety case' has already been adopted across many industries (including defence, aerospace, nuclear and railways). Studying the safety standards and guidance relating to these sectors (some of which are [1-6]), it is possible to identify a number of definitions of the safety case – some clearer than others. The definition given above attempts to cleanly define the *core* concept that is in agreement with the majority of the definitions we have discovered. The following are important aspects of the above definition:

- '**argument'** – Above all, the safety case exists to communicate an *argument*. It is used to *demonstrate* how someone can reasonably conclude that a system is acceptably safe from the evidence available.
- '**clear'** – A safety case is a device for *communicating* ideas and information, usually to a third party (e.g. a regulator). In order to do this convincingly, it must be as clear as possible.
- '**system'** – The system to which a safety case refers can be anything from a network of pipes or a software configuration to a set of operating procedures. The concept is not limited to consideration of conventional engineering 'design'.
- '**acceptably'** – Absolute safety is an unobtainable goal. Safety cases are there to convince someone that the system is safe *enough* (when compared against some definition or notion of tolerable risk).
- '**context'** – Context-free safety is impossible to argue. Almost any system can be *unsafe* if used in an inappropriate or unexpected manner. It is part of the job of the safety case to define the context within which safety is to be argued.

The safety case consists of three principal elements: Requirements, Argument and Evidence. The relationship between these three elements is depicted in Figure 1.

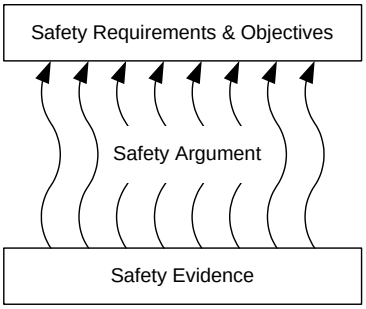

Figure 1 – The Role of Safety Argumentation

The safety argument is that which communicates the relationship between the evidence and objectives. Based on the authors' personal experience, gained from reviewing a number of safety cases, and validated through discussion with many safety practitioners, a commonly observed failing of safety cases is that the *role of the safety argument is often neglected*. In such safety cases, many pages of supporting evidence are often presented (e.g. hundreds of pages of fault trees or Failure Modes and Effects Analysis tables), but little is done to explain how this evidence relates to the safety objectives. The reader is often left to guess at an unwritten and implicit argument.

Both argument and evidence are crucial elements of the safety case that must go hand-in-hand. Argument without supporting evidence is unfounded, and therefore unconvincing. Evidence without argument is unexplained – it can be unclear that (or how) safety objectives have been satisfied. In the following section we examine how safety arguments may be clearly communicated within safety case reports. It is possible to possess a document *called* the Safety Case and for there to be no safety case (i.e. there is no compelling safety argument). In the next section we describe how safety arguments are typically communicated within any safety case report.

2. Communicating Safety Arguments

Safety arguments are most typically communicated in existing safety cases through free text. Figure 2 shows a fragment of a safety argument communicated using free text.

> The Defence in Depth principle (P65) has been addressed in this system through the provision of the following:
> - Multiple physical barriers between hazard source and the environment (see Section X)
> - A protection system to prevent breach of these barriers and to mitigate the effects of a barrier being breached (see Section Y)

Figure 2 – An Example Textual Safety Argument

In Figure 2, the text describes clearly how a safety requirement (P65) has been interpreted and achieved in the system. It also clearly provides references to where the evidence supporting the lower level statements can be found. Well-structured approaches to expressing safety arguments in text can be effective. However, there are problems experienced when text is the only medium available for expressing complex arguments. The text shown in Figure 3, taken from a real industrial safety case (with identification of the target application hidden), illustrates some of these problems.

> For hazards associated with warnings, the assumptions of [7] Section 3.4 associated with the requirement to present a warning when no equipment failure has occurred are carried forward. In particular, with respect to hazard 17 in section 5.7 [4] that for test operation, operating limits will need to be introduced to protect against the hazard, whilst further data is gathered to determine the extent of the problem.

Figure 3 – The Problems of Textual Arguments

The underlying problem of the text shown in Figure 3 is that it is unclear and poorly structured English. Not all engineers responsible for producing safety cases write clear and well-structured English. Consequently, the meaning of the text, and therefore the structure of the safety argument, can be ambiguous and unclear. Cross-references, of the type shown in Figure 3, are often necessary given the role of the safety case as an integrator of evidence. However, multiple cross-references in text can be awkward and can disrupt the flow of the main argument.

In the context of developing, agreeing, and maintaining the safety arguments within the safety case, the biggest problem with the use of free text is in ensuring that all stakeholders involved share the same understanding of the argument. Without a clear and shared understanding of the argument, safety case management is often an inefficient and ill-defined activity.

The following section describes a structured technique that has been developed to address the problems of clearly expressing and presenting safety arguments.

3. The Goal Structuring Notation

The Goal Structuring Notation (GSN) [7] – a graphical argumentation notation – explicitly represents the individual elements of any safety 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). The principal symbols of the notation are shown in Figure 4 (with example instances of each concept).

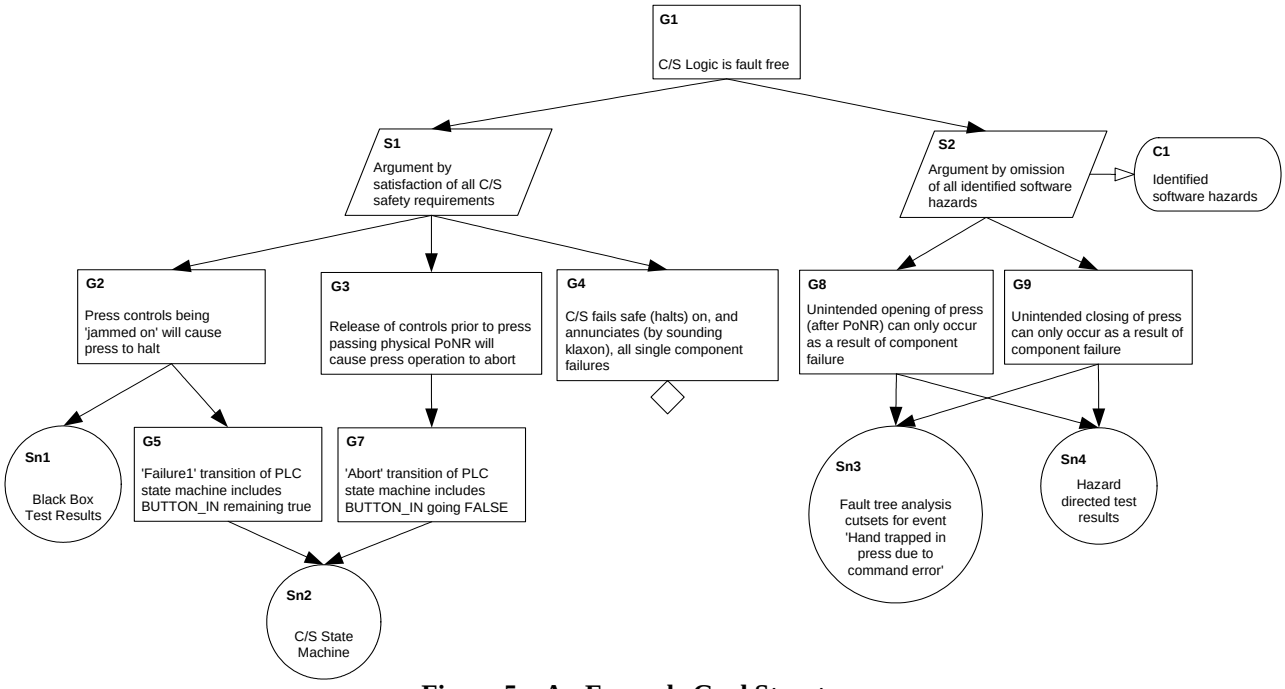

Figure 5 – An Example Goal Structure

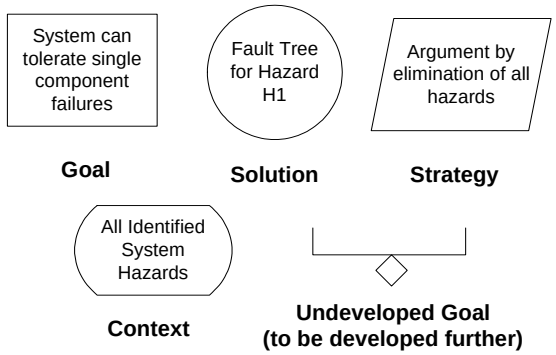

Figure 4- Principal Elements of the Goal Structuring Notation

When the elements of the GSN are linked together in a network they are described as a '*goal structure*'. The principal purpose of any goal structure is to show how goals (claims about the system) are successively broken down into sub-goals until a point is reached where claims can be supported by direct reference to available evidence (solutions). As part of this decomposition, using the GSN it is also possible to make clear the argument strategies adopted (e.g. adopting a quantitative or qualitative approach), the rationale for the approach and the context in which goals are stated (e.g. the system scope or the assumed operational role).

Figure 5 shows an example goal structure. In this structure, as in most, there exist 'top level' goals – statements that the goal structure is designed to support. In this case, "*C/S (Control System) Logic is fault free*", is the (singular) top level goal. Beneath the top level goal or goals, the structure is broken down into sub-goals, either directly or, as in this case, indirectly through a strategy. The two argument strategies put forward as a means of addressing the top level goal in Figure 5 *are "Argument by satisfaction of all C/S (Control System) safety requirements*", and, "*Argument by omission of all identified software hazards*". These strategies are then substantiated by five sub-goals. At some stage in a goal structure, a goal statement is put forward that need not be broken down and can be clearly supported by reference to some evidence. In this case, the goal "*Unintended Closing of press after PoNR (Point of No Return) can only occur as a result of component failure*", is supported by direct reference to the solutions, "*Fault tree cutsets …*" and "*Hazard Directed Testing Results*".

Within Europe, GSN has been adopted by a growing number of companies within safety-critical industries (such as aerospace, railways and defence) for the presentation of safety arguments within safety cases. The following list includes some of the applications of GSN to date:

- Eurofighter Aircraft Avionics Safety Justification
- Hawk Aircraft Safety Justification
- U.K. Ministry of Defence Site Safety Justifications

- U.K. Dorset Coast Railway Re-signalling Safety Justification
- Submarine Propulsion Safety Justifications
- Safety Justification of UK Military Air Traffic Management Systems
- London Underground Jubilee Line Extension Safety Justification
- Swedish Air Traffic Control Applications
- Rolls-Royce Trent Engine Control Systems Safety Arguments

Published results of industrial experience of using GSN can be found in [8]. The key benefit experienced by those companies adopting GSN is that it improves the comprehension of the safety argument amongst all of the key project stakeholders (i.e. system developers, safety engineers, independent assessors and certification authorities). In turn, this has improved the quality of the debate and discussion amongst the stakeholders and has reduced the time taken to reach agreement on the argument approaches being adopted. However, having a clear means of communicating safety arguments is only a partial answer to the challenge of establishing a systematic safety case development approach. In addition, it is important to consider the *timing* of safety case development with respect to the system development lifecycle. The use of GSN to facilitate an evolutionary and systematic approach to safety case construction, in step with system development, is discussed in [9].

4. Extensions to GSN

In addition to the "vanilla" Goal Structuring Notation described in this paper there are a number of extensions to GSN which have been developed to aid and improve industrial application of the technique. These extensions are introduced briefly in the following sub-sections.

4.1 Maintenance of Safety Arguments

A crucial aspect of safety case management is the ongoing maintenance of the safety argument through life. Throughout the operational life of any system, changing regulatory requirements, additional safety evidence and a changing design can challenge the corresponding safety case. In order to maintain an accurate account of the safety of the system, all such challenges must be assessed for their impact on the original safety argument. This is increasingly being recognised by many safety standards. However, many safety engineers are experiencing difficulties with safety case maintenance at present, the prime reason being that they do not have a systematic and methodical approach by which to examine the impact of change on safety argument. The size and complexity of safety arguments and evidence being presented within safety cases is increasing. In [10] a process, based upon GSN, is defined and described which attempts to address these difficulties through facilitating the systematic impact assessment of safety case challenges.

4.2 Safety Case Patterns

Common structures in safety case arguments can be reused through their documentation as 'Safety Case Patterns'. This approach can circumvent some of the problems with the existing, informal and ad-hoc approaches to safety case material reuse. Through the explicit capturing and documentation of reusable safety case elements as patterns, the process of safety case construction and reuse can be made more systematic. In [11] a description of a safety case pattern language based on the Goal Structuring Notation is presented. Similarly, Safety Case AntiPatterns [7 & 12] can be used to communicate weak and flawed safety arguments, such that they may be recognised and avoided in future developments.

4.2.1. Software Safety Case Patterns. The guidance found in most standards for the development of software for safety critical systems identifies processes for different safety integrity levels (SILs). Software is shown to be fit for use primarily by appeal to the standards, supported with appropriate evidence. The assumption is that software developed against the requirements of higher SILs will be less prone to critical failures. This assumption has been questioned [13], and instead it has been proposed that an "evidence-based" approach be taken to software [14]. To implement this type of approach requires arguments to reflect the contribution of software to safety in the context of the system. An "evidence-based" approach has been implemented in [12] by using a framework for articulating software safety arguments, based on categorisation of evidence, which is largely independent of the development process. The safety case pattern language has been used to develop a catalogue of patterns to describe this framework.

4.3. Modular Safety Cases

The adoption of Integrated Modular Avionics (IMA) in the aerospace industry offers potential benefits of improved flexibility in function allocation, reduced development costs and improved maintainability. However, it requires a new

certification approach. The traditional approach to certification is to prepare monolithic safety cases as bespoke developments for a specific system in a fixed configuration. However, this nullifies the benefits of flexibility and reduced rework claimed of IMA-based systems and will necessitate the development of new safety cases for all possible (current and future) configurations of the architecture. A modular approach to safety case construction [15], based upon GSN, allows the safety case to be partitioned into separable arguments of safety corresponding with the components of the system architecture. This is applicable to IMA and other modular based systems where certification of different components or modules may occur at different stages.

4.4 Assurance of Safety Arguments

Implicit in the assessment of a safety case is a consideration of whether the safety argument 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. In this sub-section we present ongoing research into an approach for considering and explicitly describing safety case assurance [16 & 12]. The approach described is a process which occurs in the development and assessment of a safety case but currently remains unexpressed.

The safety argument claims made in 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 "failure rate of component X is 10^{-4} failures per operational hour" is either a true or false. 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 – if premises are true, conclusion is true – and Sound – an argument which is valid and has true premises.

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, GSN accepts arguments that are consistent – if premises are true, conclusion may be true – and thus causally weaker.

This weaker form of causal relationship is known as *inductive* argumentation – the conclusion follows from the premises not with necessity but only with probability. While the stronger, valid argument form is known as *deductive* argumentation – if premises are true, then the conclusion must also be true.

The inductive nature of GSN safety arguments 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.

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.

The term 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. We defined Safety Assurance as: *A qualitative statement expressing the degree of confidence that a safety claim is true.*

The size and complexity of safety arguments combined with the subjective nature of argument composition is such that assurance cannot easily be considered quantitatively. 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

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

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 can be used. An approach for implementing Safety Assurance Levels is described in [12] and [16]. This approach can be used to express the assurance provided by an argument based upon the type of support provided by the argument and evidence.

5. Conclusion

In this paper we have described the safety case concept as adopted by many safety critical industries (such as defence, railways and aerospace) within Europe. The principal objective of a safety case is to present an argument that a system is acceptably safe to operate in a given context. However, the safety *argument* is often poorly communicated through the textual narrative of safety case reports. The Goal Structuring Notation (GSN), presented within this paper, has been developed to provide a clear, structured, approach to developing and presenting safety arguments. It has been widely adopted across a number of safety critical industries for development of safety cases. The adoption of GSN as a structured argumentation technique has allowed users to consider advanced concepts such as patterns of argument, argument maintenance and managing levels of argument assurance. Increasingly, this success of GSN within the safety domain has prompted consideration of its wider use in other domains where assurance cases are also required, such as security.

6. References

[1] U.K. Health and Safety Executive, *A guide to the Offshore Installations (Safety Case) Regulations 1992*, Health and Safety Executive, HSE Books 1992.

[2] U.K. Health and Safety Executive, *Railway Safety Cases - Railway (Safety Case) Regulations 1994 - Guidance on Regulations*, Health and Safety Executive, HSE Books 1994.

[3] Railtrack, *The Yellow Book: Engineering Safety Management*, Railtrack, 2000.

[4] U.K. Ministry of Defence, *JSP 430 - Ship Safety Management System Handbook*, Ministry of Defence, January 1996.

[5] U.K. Ministry of Defence, *00-56 Safety Management Requirements for Defence Systems*, Ministry of Defence, Defence Standard, December 1996.

[6] Nuclear Safety Directorate, *Nuclear Site License Conditions*, Nuclear Safety Directorate, Health and Safety Executive, 2000.

[7] T. P. Kelly, *Arguing Safety – A Systematic Approach to Safety Case Management*, DPhil Thesis YCST99-05, Department of Computer Science, University of York, UK, 1998.

[8] P. Chinneck, D.J. Pumfrey & T.P. Kelly, "Turning up the HEAT on Safety Case Construction", in *Practical Elements of Safety: Proceedings of the Twelfth Safety-critical Systems Symposium*, Ed. F. Redmill, & T. Anderson, Springer, Birmingham, UK, 2004, pp. 223-240.

[9] T P Kelly, "A Systematic Approach to Safety Case Management", in CAE Methods for Vehicle Crash Worthiness and Occupant Safety, and safety critical systems, *SAE 2004 World Congress Special Publication SP-1870*, Society of Automated Engineers, 2004.

[10] T P Kelly, J A McDermid, "A Systematic Approach to Safety Case Maintenance", *Reliability Engineering and System Safety* vol. 71, Elsevier, 2001, pp271-284.

[11] T P Kelly & J A McDermid, "Safety Case Construction and Reuse using Patterns", in *Proceedings of 16th International Conference on Computer Safety, Reliability and Security* (SAFECOMP'97), Springer-Verlag, September 1997.

[12] R. A. Weaver, *The Safety of Software – Constructing and Assuring Arguments*, DPhil Thesis, Department of Computer Science, University of York, UK, 2003.

[13] J. A. McDermid and D. J. Pumfrey, "Software Safety: Why is there no Consensus?", in *Proceedings of the 19th International System Safety Conference* (ISSC 2001), System Safety Society, Huntsville, USA, 2001.

[14] R A Weaver, J McDermid, "T P Kelly Software Safety Arguments: Towards a Systematic Categorisation of Evidence" in *Proceedings of the 20th International System Safety Conference* (ISSC 2002), System Safety Society, Denver, Colorado, USA, 2002.

[15] I J Bate & T P Kelly, "Architectural Considerations in the Certification of Modular Systems", in *Proceedings of the 21st International Conference on Computer Safety, Reliability and Security* (SAFECOMP'02), Springer-Verlag, September 2002.

[16] R. A. Weaver, J. Fenn, T. P. Kelly, "A Pragmatic Approach to Reasoning about the Assurance of Safety Arguments" in *Proceedings of 8th Australian Workshop on Safety Critical Systems and Software (SCS'03)*, Canberra, Australia 2003. Published in *Conferences in Research and Practice in Information Technology Series*, P. Lindsay and T. Cant (Eds.), vol.33, Australian Computer Society, 2003.