SAFETY CASE NOTATIONS: ALTERNATIVES FOR THE NON-GRAPHICALLY INCLINED?

C.M. Holloway

NASA Langley Research Center, Hampton, Virginia , USA; c.m.holloway@nasa.gov

Keywords: safety case, visual, verbal, notations, clarity.

Abstract

This working paper presents preliminary ideas of five possible text-based notations for representing safety cases, which may be easier for non-graphically inclined people to use and understand than the currently popular graphics-based representations.

1 Introduction

One does not have to subscribe to the complete tenets of any particular theory of learning styles to recognize that some people prefer visual presentation of information and other people prefer verbal presentation. For the former, "a picture is worth a thousand pictures" [17]; but for the later, "a word is worth a thousand pictures" [3].

Currently, the most well-known notations for describing safety cases are graphics-based. The Goal-Structuring Notation (GSN) is a prototypical example of such a notation [12]. It uses a small set of graphical elements, annotated with text, and connected by directed lines to represent the elements of a safety argument and the relationships among these elements. Advocates of GSN (and similar notations) assert that graphical representations simplify the construction and managing of safety arguments and facilitate the presentation of these arguments to others (see for example [4, 20, 22]).

These assertions are almost certainly true for those members of the population who are predisposed to a visual learning style. For such people, a picture *is* worth a thousand words. They are likely to find a GSN representation of an argument easy to create and understand, particularly when compared to a non-graphical representation of the same argument.

But, what about those people who do not have a visual learning style; those who prefer words to pictures? Such people seem likely to find a GSN representation of an argument confusing, or at the very least, unappealing, and may struggle to create these representations of their own arguments. This is certainly true for me.

In this paper, I sketch five styles of text-based representations for safety arguments, and conclude with brief, preliminary, subjective observations about possible next steps to take.

2 Background

This section provides a brief introduction to safety cases and to GSN. Readers already familiar with these concepts may wish to skip to the next section.

2.1 Safety cases

For the development and certification of safety critical software systems, a move away from process-based standards towards evidence-based standards has begun in some countries, and is strongly advocated in others [10]. Instead of requiring adherence to particular constraints on the process by which a system is developed, an evidence-based approach requires the creation of a *safety case* for the system. A safety case consists of explicit safety requirements, the evidence that the requirements have been met, and the argument linking the evidence to the requirements. Both the argument and the evidence are essential. An argument without adequate supporting evidence is, or at least should be, unconvincing. A body of evidence without an argument is unexplained. In both cases, knowing whether the system's safety requirements have been met is difficult [21].

2.2 Goal-Structuring Notation

This paper uses GSN as the example graphics-based notation for expressing safety cases. Some of the primary symbols of the notation are illustrated in figure 1, and explained below [6]:

- A *goal* states a claim (or, for those who prefer different words, proposition or statement) that is to be established by an argument. A GSN diagram (called a goal structure) will usually have a top-level goal, which will often be decomposed into more goals.
- A *strategy* describes the method used to decompose a goal into additional goals.
- A *solution* describes the evidence that a goal has been met.
- The *context* associated with another GSN element lists information that is relevant to that element. For example, the context of a particular goal might provide definitions necessary to understand the meaning of the goal.
- An *assumption* is a statement that is taken to be true, without further argument or explanation.

- A *justification* explains why a solution provides sufficient evidence to satisfy a goal.

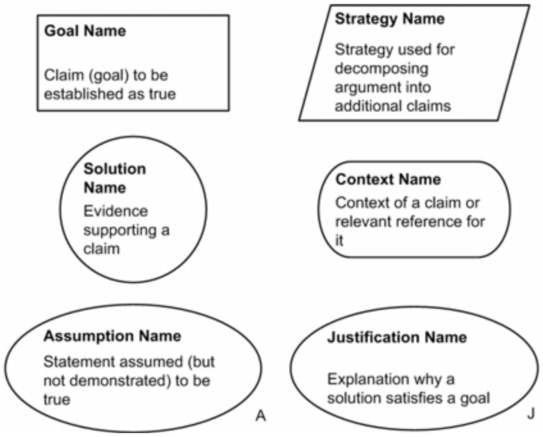
Goal Name
Claim (goal) to be established as true
Strategy Name
Strategy used for decomposing argument into additional claims
Solution Name
Evidence supporting a claim
Context Name
Context of a claim or relevant reference for it
Assumption Name
Statement assumed (but not demonstrated) to be true
A
Justification Name
Explanation why a solution satisfies a goal
J

Figure 1: Some elements of GSN

To construct an argument, the elements of the GSN notation are linked together using directed lines. An example of a safety argument goal structure is shown in figure 2 (based on [14]).

This goal structure is necessarily fairly simple; however it does contain sufficient detail and variety to serve as the basis for showing what a safety case argument might look like in each of the text-based notations described in the next section.

3 Text-based Notations

In this section five text-based notations for expressing arguments are described.

3.1 Normal Prose

Law and philosophy are two disciplines that are particularly concerned with the construction and expression of coherent, cogent arguments. In both disciplines, normal prose is the most common medium of expression (as examples see [1, 8, 9, 13]). Although the use of graphical notations has been researched in these fields, and it continues to be researched [2], such notations have generally not found much favour. Court opinions, legal briefs, law review articles, and law books consist almost entirely of prose, albeit sometimes rather bland, and heavily footnoted prose. Philosophical writings are similar. Except for the occasional diagram, one usually encounters nothing more than normal text.

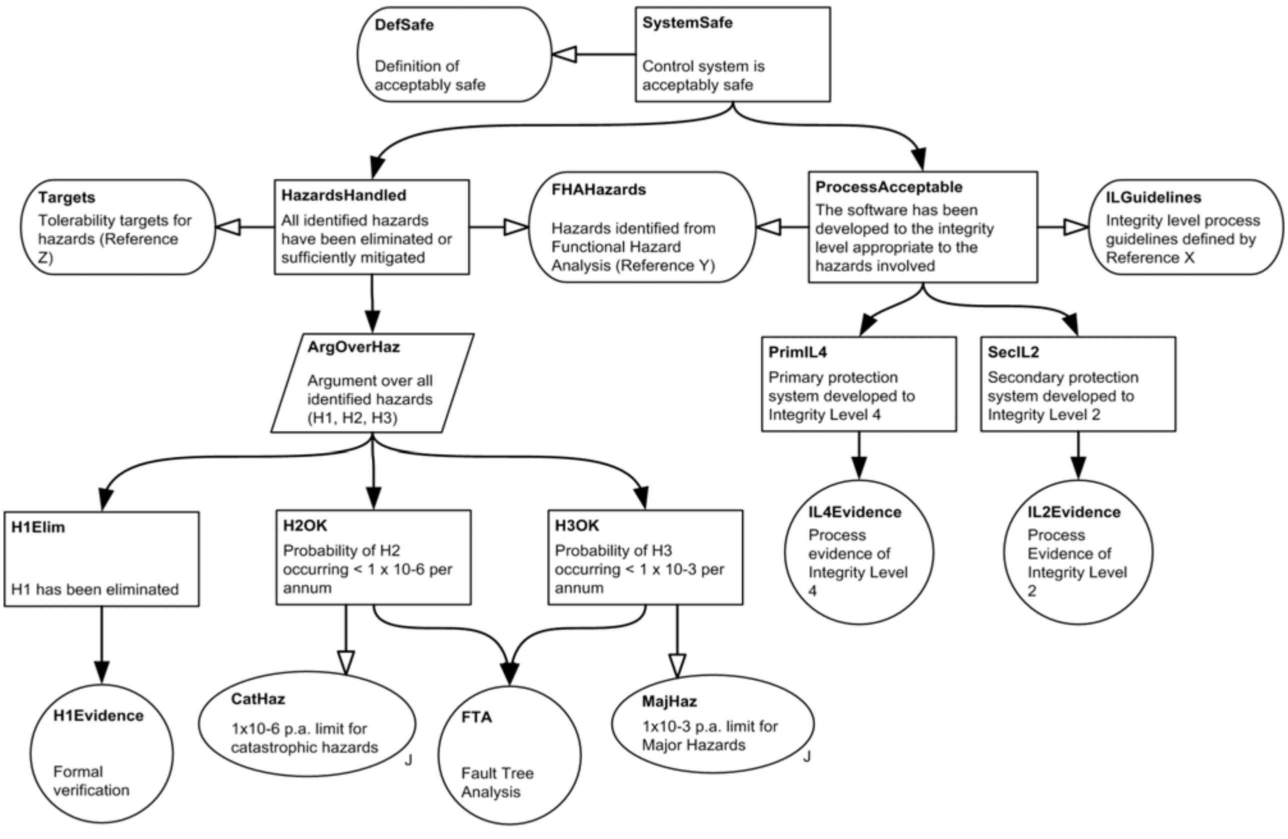
DefSafe
Definition of acceptably safe
SystemSafe
Control system is acceptably safe
Targets
Tolerability targets for hazards (Reference Z)
HazardsHandled
All identified hazards have been eliminated or sufficiently mitigated
FHAHazards
Hazards identified from Functional Hazard Analysis (Reference Y)
ProcessAcceptable
The software has been developed to the integrity level appropriate to the hazards involved
ILGuidelines
Integrity level process guidelines defined by Reference X
ArgOverHaz
Argument over all identified hazards (H1, H2, H3)
PrimIL4
Primary protection system developed to Integrity Level 4
SecIL2
Secondary protection system developed to Integrity Level 2
H1Elim
H1 has been eliminated
H2OK
Probability of H2 occurring &lt; 1 x 10-6 per annum
H3OK
Probability of H3 occurring &lt; 1 x 10-3 per annum
IL4Evidence
Process evidence of Integrity Level 4
IL2Evidence
Process Evidence of Integrity Level 2
H1Evidence
Formal verification
CatHaz
1x10-6 p.a. limit for catastrophic hazards
J
FTA
Fault Tree Analysis
MajHaz
1x10-3 p.a. limit for Major Hazards
J

Figure 2: Example Goal Structure

Safety case arguments may also be written in normal prose. Shown below is one way that the safety argument of figure 2 may be expressed in prose.

The control system is acceptably safe, given a definition of acceptably safe, because all identified hazards have been eliminated or sufficiently mitigated and the software has been developed to the integrity levels appropriate to the hazards involved.

Given both the tolerability targets for hazards (from reference Z), and the list of hazards identified from the functional hazard analysis (from reference Y), we can show that all identified hazards have been identified or sufficiently mitigated by arguing over all three of the identified hazards: H1, H2, and H3.

We know from the formal verification we conducted that H1 has been eliminated.

We know that catastrophic hazard H2 has been sufficiently mitigated because fault tree analysis shows that its probability of occurrence is less than $1x10^{-6}$ per annum, and the acceptable probability in our environment for a catastrophic hazard is $1x10^{-6}$ per annum.

We know that major hazard H3 has been sufficiently mitigated because fault tree analysis shows that its probability of occurrence is the less than the $1x10^{-3}$ per annum permitted for the occurrence of a major hazard.

Given the list of hazards identified from the functional hazard analysis in reference Y, and the integrity level (IL) process guidelines defined in reference X, we can show that the software has been developed to the integrity level appropriate to the hazards involved. First, the process evidence for the primary protection system shows that it was developed to the required IL 4. Second, the process evidence for the secondary protection system shows that it was developed to the required IL 2.

3.2 Structured Prose

One common criticism, among several, of prose safety arguments is that the structure of the argument may be lost among all the words[1]. One way to address this criticism is add some explicit structure to the prose. A structured prose presentation of an argument reduces the freedom of expression possible in normal prose by requiring that the critical parts of the argument be explicitly denoted.

The example below uses slightly modified GSN-inspired terminology for denotation purposes ('claim' instead of 'goal', and 'evidence' instead of 'solution').

This argument establishes the following **claim**: the control system is acceptably safe, within the **context** of a definition of acceptably safe. To establish the top-level claim, two **sub-claims** are established: (1) all identified hazards have been eliminated or sufficiently mitigated and (2) the software has been developed to the integrity levels appropriate to the hazards involved.

Within the **context** of the tolerability targets for hazards (from reference Z) and the list of hazards identified from the functional hazard analysis (from reference Y), we follow the **strategy** of arguing over all three of the identified hazards (H1, H2, and H3) to establish sub-claim 1, yielding three additional **claims**: H1 has been eliminated; H2 has been sufficiently mitigated; and H3 has been sufficiently mitigated.

The **evidence** that H1 has been eliminated is formal verification.

The **evidence** that catastrophic hazard H2 has been sufficiently mitigated is a fault tree analysis showing that its probability of occurrence is less than $1x10^{-6}$ per annum. The **justification** for using this evidence is that the acceptable probability in our environment for a catastrophic hazard is $1x10^{-6}$ per annum.

The **evidence** that the major hazard H3 has been sufficiently mitigated is a fault tree analysis showing that its probability of occurrence is less than $1x10^{-3}$ per annum. The **justification** for using this evidence is that the acceptable probability in our environment for a major hazard is $1x10^{-3}$ per annum.

We establish sub-claim (2) within the **context** of the list of hazards identified from the functional hazard analysis in reference Y, and the integrity level (IL) process guidelines defined in reference X. The process **evidence** shows that the primary protection system was developed to the required IL 4. The process **evidence** also shows that the secondary protection system was developed to the required IL 2.

3.3 Argument Outline

The structure of the argument may be made even more explicit by adopting an outline format. With such a format, simple declarative language is used, similar to the text contained within a GSN element. Indentation, numbering, and font changes may also be used to help emphasize the overall argument structure. As with outlines in general, several different specific formats are possible. The format

[1] Without a doubt it is possible to obscure an argument within prose; however, that possibility does not have to be realized. In previous work, the most cogent, easy to understand safety case was presented in prose [7].

chosen for the example below is similar but not identical to the format suggested in [16].

Claim 1: Control system is acceptably safe.
Context 1: Definition of acceptably safe.

Claim 1.1: All identified hazards have been eliminated or sufficiently mitigated.
Context 1.1-a: Tolerability targets for hazards (reference Z).
Context 1.1-b: Hazards identified from functional hazard analysis (reference Y).

Strategy 1.1: Argument over all identified hazards (H1, H2, H3)

Claim 1.1.1: H1 has been eliminated.
Evidence 1.1.1: Formal verification

Claim 1.1.2: Probability of H2 occurring < $1x10^{-6}$ per annum.
Justification 1.1.2: $1x10^{-6}$ per annum limit for catastrophic hazards.
Evidence 1.1.2.: Fault Tree analysis.

Claim 1.1.3: Probability of H3 occurring < $1x10^{-3}$ per annum.
Justification 1.1.3: $1x10^{-3}$ per annum limit for major hazards.
Evidence 1.1.3: Fault tree analysis.

Claim 1.2: The software has been developed to the integrity level appropriate to the hazards involved.
Context 1.2-a: (same as Context 1.1-b)
Context 1.2-b: Integrity level (IL) process guidelines defined by reference X.

Claim 1.2.1: Primary protection system developed to IL 4.
Evidence 1.2.1: Process evidence of IL 4

Claim 1.2.2: Secondary protection system developed to IL 2.
Evidence 1.2.2: Process evidence of IL 2.

3.4 Mathematical Proof

Although geometric proofs are the bane of many secondary school students, the basic format employed in them is appropriate for consideration as a format for expressing safety case arguments. The following instantiation of the argument from figure 2 shows one way in which this could be done. There are two departures from the typical proof approach. The first departure is that the word 'establish' is used instead of 'prove'. This is because safety arguments are rarely, if ever, deductive (and thus have conclusions that may be proved in the sense used in geometry); instead the arguments are inductive (and thus have conclusions that may only be established to some level of confidence) [5]. The second departure is that the top-down nature of the safety argument is maintained by permitting references to statements established later.

Establish:
SystemSafe: Control system is acceptably safe

Given:
A. Definition of acceptably safe.

Statements	*Reasons*
1. All identified hazards have been eliminated or sufficiently mitigated	1. HazardsHandled (established below)
2. The software has been developed to the integrity level appropriate to the hazards identified	2. ProcessAcceptable (established below)
3. Control system is acceptably safe	3. 1,2

Establish:
HazardsHandled: All identified hazards have been eliminated or sufficiently mitigated

Given:
A. Tolerability targets for hazards (reference Z).
B. Hazards identified from functional hazard analysis (reference Y).

By: Arguing over all identified hazards (H1, H2, H3)

Statements	*Reasons*
1. H1 has been eliminated	1. formal verification
2. p(H2) < $1x10^{-6}$ per annum (p.a.)	2. fault tree analysis
3. Upper limit on permitted catastrophic hazard occurrence is $1x10^{-6}$ p.a.	3. Given (A)
4. H2 has been mitigated	4. 2, 3
5. p(H3) < $1x10^{-3}$ p. a.	5. fault tree analysis
6. Upper limit on permitted probability of a major hazard occurrence is $1x10^{-3}$ p.a.	6. Given (A)
7. H3 has been mitigated	7. 5, 6
8. All identified hazards have been eliminated or sufficiently mitigated	8. Given (B), 4, 7

Establish:
ProcessAcceptable: The software has been developed to the integrity level appropriate to the hazards identified.

Given:
A. Integrity level process guidelines defined by reference X.
B. Hazards identified from functional hazard analysis (reference Y).

Statements	*Reasons*
1. Primary protection system has hazards requiring development to integrity level 4	1. Given (A, B)
2. Primary protection system developed to integrity level 4	2. Process evidence of integrity level 4
3. Primary protection system developed to appropriate level	3. 1, 2
4. Secondary protection has hazards requiring development to integrity level 2	4. Given (A, B)
5. Secondary protection developed to integrity level 2	5. Process evidence of integrity level 2
6. Secondary protection system developed to appropriate level	6. 4, 5
7. The software has been developed to the integrity level appropriate to the hazards identified	7. 3, 6

3.5 LISP Style

The fifth text-based representation for safety case arguments is one based on the programming language LISP [18]. LISP-based notations have been employed in a wide range of applications over the years, including the theorem proving system ACL2 (A Computational Logic for Applicative Common LISP) [11].

The example below presents one way of describing the argument of figure 2 using a LISP style notation. Note that that short names, rather than full text, are used for the argument contents. The standard names used throughout this paper for the GSN element types also used (claim, context, etc.). In analogy to functions or operators in a LISP program, these appear at the beginning of a parenthesized list, and should be thought of as applying to everything that follows within that list.

```
(context DefSafe
  (claim SystemSafe
     (context FHAHazards
        (context Targets
           (claim HazardsHandled
              (strategy ArgOveHaz
                 (claim H1Elim
                    (evidence H1Evidence))
                 (claim H2OK
                    (justification CatHaz)
                    (evidence FTA))
                 (claim H3OK
                    (justification MajHaz)
                    (evidence FTA)))))
        (context ILGuidelines
           (claim ProcessAcceptable
              (claim PrimIL4
                 (evidence IL4Evidence))
              (claim SecIL2
                 (evidence IL2Evidence)))))))
```

4 Concluding Remarks

My work to date in this area has consisted in creating the five examples presented above, with specific concentration on developing the mathematical proof and LISP style representations. Careful analysis and evaluation of the efficacy of these representations has not been done, but I will conclude this paper with two observations and an opinion based on the work completed so far.

One observation is that any evaluation of a notation for representing safety arguments must consider a host of questions, including the following:

- Which is worse: an argument that *fails to convince* others that a claim is satisfied *when it is satisfied*, or an argument that *falsely convinces* others that a claim is satisfied *when it is not*?
- Which is most important: ease of argument creation, perspicuity of a created argument, manual analysis of an argument, automated analysis of an argument, or ease of argument maintenance?
- Who is the intended audience for the argument?
- How is the intended audience distributed in regards to preferences for visual or verbal information?
- If the notation is primarily intended for visual people, in what ways does it accommodate verbal people? And vice versa?

The other observation is that devising an experiment that would take into account these questions to evaluate adequately the benefits and disadvantages of several different representations for safety arguments is a very difficult task. Which probably explains why it has apparently not been done yet.

The opinion is that eventually the safety community will settle on some form of text-based notation as the primary means for creating, analyzing, and manipulating safety-cases, with graphical representations serving only an auxiliary role. Only time will tell if this opinion turns out to be right, but history in philosophical and legal argument is on its side.

References

[1] G. Bahnsen. "A Conditional Resolution of the Apparent Paradox of Self-Deception", Ph.D. dissertation, University of Southern California (1978).

[2] Cardozo Law School. Conference on Graphic and Visual Representations of Evidence and Inference in Legal Settings, New York, New York, USA (2007). [papers and presentations available at <http://tillers.net/conference.html>, visited 19 June 2008].

[3] J. Doumont. "Verbal Versus Visual: A Word Is Worth a Thousand Pictures, Too", *Technical Communication*, Volume 49, Number 2, pp. 219-224, (2002).

[4] L. Emmet, G. Cleland. "Graphical Notations, Narratives and Persuasion: a Pliant Systems approach to Hypertext Tool Design", *Proceedings of the 13th ACM Conference on Hypertext and Hypermedia*, College Park, Maryland, USA, (2002).

[5] Trudy Govier. *A Practical Study of Argument*, sixth edition, Wadsworth/Thomson Learning, Belmont, California, USA, (2005).

[6] W. S. Greenwell. "Pandora: An Approach to Analyzing Safety-Related Digital-System Failures", Ph.D. dissertation, University of Virginia (2007).

[7] W. S. Greenwell, J. C. Knight, C. M. Holloway, J. J. Pease. "A Taxonomy of Fallacies in System Safety Arguments", *Proceedings of the 24th International System Safety Conference*, Albuquerque, New Mexico, USA, (2006).

[8] H. L. A. Hart, Tony Honoré. *Causation in the Law*, second edition, Clarendon Press, Oxford, UK (1985).

[9] D. Hume. *A Treatise of Human Nature*, edited by D. F. Norton, M. J. Norton, Oxford University Press, Oxford, UK (2000).

[10] D. Jackson, M. Thomas, L. I. Millett (eds). Software for Dependable Systems: Sufficient Evidence? National Research Council, Committee on Certifiably Dependable Software Systems, (2007). Available at <http://www.nap.edu/catalog/11923.html> [visited 8 May 2008].

[11] M. Kaufmann, J. S. Moore. "ACL2 Version 3.3", <http://www.cs.utexas.edu/~moore/acl2> (2007). [visited 7 August 2008].

[12] T. P. Kelly. "Arguing Safety --- A Systematic Approach to Safety Case Management", DPhil Thesis, Department of Computer Science, University of York, UK (1998).

[13] *McConnell v. Federal Election Commission*, 540 U.S. 93 (2003).

[14] J. McDermid. "Enablers for the Assurance Case for Safety, Security and Dependability", presented at the 8th Semi-Annual Software Assurance Forum, Gaithersburg, Maryland, USA, (2008). Available at <https://buildsecurityin.us-cert.ogv/swa/forum_May_2008.html> [visited 21 July 2008].

[15] E. A. Nguyen, W. S. Greenwell, M. J. Hecht. "Using an Assurance Case to Support Independent Assessment of the Transition to a New GPS Ground Control System", *Proceedings of the International Conference on Dependable Systems & Networks*, Anchorage, Alaska, 24-27 June (2008).

[16] Software Engineering Institute. "Assurance Case and Plan Preparation", <http://www.sei.cmu.edu/pcs/acprep.html> [visited 23 July 2008].

[17] J. Speake (ed.). "One PICTURE is worth ten thousand words", *The Oxford Dictionary of Proverbs*, Oxford University Press (2003). [retrieved via University of Virginia Library at Oxford Reference Online <http://www.oxfordreference.com/views/ENTRY.html?subview=Main&entry=t90.e1627>, 4 August 2008].

[18] G. L. Steele. Common Lisp: The Language, Digital Press, Bedford, Massachusetts, USA, (1990).

[19] S. E. Toulmin. *The Uses of Argument*, updated edition, Cambridge University Press, (2003).

[20] R. A. Weaver, T. P. Kelly. "The Goal Structuring Notation: A Safety Argument Notation", *Proceedings of Dependable Systems & Networks Workshop on Assurance Cases*, (2004).

[21] R. A. Weaver, G. Despotou, T. P. Kelly, J. McDermid. "Combining Software Evidence: Arguments and Assurance", Proceedings of ICSE 2005: Workshop on Realising Evidence Based Software Engineering, St. Louis, Missouri, USA, (2005).

[22] R. A. Weaver. "The Safety of Software --- Constructing and Assuring Arguments", DPhil Thesis, Department of Computer Science, University of York, UK (2003).