Experience with Assurance Case Preparation

Chris Hobbs[1]

Abstract
This short paper describes how Assurance Case preparation has recently changed within *BlackBerry QNX*.

[1] *chobbs@qnx.com*

1. Background

In §5.3.1 of ISO 26262-10 (reference [8]) it is stated that

> *There are three principal elements of a safety case, namely: the requirements; the argument; and the evidence.*[1]

This document addresses only the question of presenting the argument and, in particular, the notation used therefor.

2. GSN and BBN

The Goal Structuring Notation (GSN) (reference [3]) was devised for expressing an Assurance Case argument, but has a number of limitations:

1. **It has limited mechanisms for expressing doubt.** A claim (rectangle) with a yellow background is available as an extension to the GSN and *"may be used to indicate that there is counter-evidence which casts doubt on the goal's validity"*, but there is no mechanism within the notation for handling the implications of this.

   Quite often there is some level of doubt associated with an argument. For example, in order to demonstrate that the programmers working on a particular development are competent (see § 5.4.3 of ISO 26262-2) a claim might be made that all have completed training courses, the associated evidence being their training records. But there might be doubt about the quality, relevance and recency of the training courses: that training was provided by a teacher presenting the material for the first time, it concentrated on C rather than C++ and was taken 2 years ago. The fact that the team has been trained is incontrovertible, the quality of the training is what is in doubt.

2. **It does not allow different emphasis to be placed on different pieces of evidence** ("solutions" in GSN terms). It may be useful, for example, when performing a sensitivity analysis, to express the idea that while three pieces of evidence are being presented to justify a certain claim, evidence *A* is the one on which most reliance is being placed, with *B* and *C* as supporting evidence.

3. **It is not quantitative.** This was a deliberate choice in the design of GSN, it apparently being felt that quantification is always to some extent a fiction, being based on guesses and "engineering estimates". From rough input estimates ("about 60%", "about 75%"), one can get an output confidence value of 78.73662772635% in the overall argument and this spurious precision can mislead. However lack of quantification further reduces the possibility of performing sensitivity analysis.

There is also some evidence (see reference [6]) that the use of GSN leads to confirmation bias. With GSN, the analyst sets out to find an argument that the system under consideration **is** safe. As illustrated by the Nimrod report (reference [5]), this can be a very dangerous attitude when producing an Assurance Case.

To address this, *BlackBerry QNX* has traditionally prepared two Assurance Case arguments:

1. A GSN expression of the argument. This has been prepared because it represents the format expected and accepted by external auditors.

2. A Bayesian Belief Network (BBN) expression of the argument. This has been prepared to convince *BlackBerry QNX's* own Safety Engineers of the validity of the argument. This approach has been described in numerous papers including references [9], [4] and [7].

   The BBN notation allows doubt to be expressed, allows different stress to be placed on different sub-arguments and is quantitative.

[1] Punctuation as in the original.

3. New Approach

BlackBerry QNX has augmented GSN with “Eliminative Induction” semantics, as described in reference [2]. The name has subsequently changed to “Eliminative Argumentation” (reference [1]), but the concept is still much the same.

Using eliminative induction (or eliminative argumentation) has changed the face of GSN completely, allowing a harsh, cold light to be shone into the corners of the argument and exposing any nasty fauna lurking there.

It effectively turns the task of the engineer producing the Assurance Case around — encouraging her to record all doubts about the safety of the system under consideration. This uses confirmation bias positively by asking the engineer to argue that the system is unsafe. It should then be possible to eliminate all of those arguments.

In particular, three types of doubt are encouraged: rebutting doubts (the claim is wrong!), undermining doubts (the evidence is wrong!) and undercutting doubts (the claim may be correct and the evidence may be convincing, but the chain from the claim to the evidence is weak).

The approach is still not quantitative, but it provides the ability to doubt, not only the validity and relevance of a piece of evidence, but also the structure of the argument itself.

Of course, shining lights into dark corners may not be good when trying to push an inadequate product through certification, but it certainly appears to ensure that the Assurance Case is more honest. And that is a good thing when safety is concerned.

4. Results

QNX prepared Assurance Cases for its operating system product in 2010, 2012, 2013, 2014, 2015, 2016, in each using both GSN and Bayesian Belief Network. In each of these, the Assurance Case was professionally prepared and satisfied auditors for the issuing of certificates against IEC 61508 at SIL 3 and ISO 26262 at ASIL D (2013 and later).

For its 2018 certification of the same product, QNX made use of the eliminative induction technique described above.

Without the need for a Bayesian Belief Network, this latest Assurance Case uncovered more than twenty previously missed problems: some procedural and some technical.

Some problems were trivial and could be fixed immediately, others required process changes that could only be implemented over time and these latter problems led to thirteen Concession Requests being submitted to the certification body (TÜV Rheinland). These Concession Requests effectively say, *“We have identified a problem with our adherence to paragraph X of ISO 26262, but cannot fix it immediately. We have put together the following plan to resolve the problem and will provide regular updates on how the remedial action is progressing. . . ”*.

5. Examples

As described above, more than twenty previously-missed problems were found by using eliminative induction. This section describes a few of them. Note that all of the examples given here could have been found during audits without the use of eliminative induction. However, they were not.

5.1 Example: Static Analysis Checks

QNX’s processes require that, when a programmer puts a code change out for review, he or she must also publish the results of a static code analysis check. If the static analysis tool has issued a new warning, then this must be declared and justified. An additional approval then has to be obtained before the code is checked into the repository.

During the preparation of the GSN and BBN arguments, the question was always raised as to whether this process was being followed. The response from the programmers and from an audit of some random review requests indicated that it was.

When the question was inverted and became “can you think of any occasion when the process was not followed”, some programmers said that there was something that had concerned them for some time. The static analysis results were always being published when the code was first put up for review. However, if the review required the code to be changed and re-submitted (sometimes several times), the static analysis results were not always being republished on the subsequent iterations.

In itself, this might not be considered an important matter, but it was interesting how the programmers who reported it had been concerned about it, but had really had no way to report their discomfort. Had there been a serious safety issue, this would doubtlessly have been reported as part of QNX’s safety culture, but a niggling concern like this was not serious enough to report.

5.2 Example: Bug Report Publication

It is important that all customers of QNX’s certified operating system be aware of any bugs that are found in it that could possibly affect safety. QNX carries out an Impact Analysis on all reported bugs (whether found by customers or in the test laboratory) and, where there is a possibility of safety being affected, details of the bug and the work-around are published in a document that is circulated to customers periodically.

During the analysis using eliminative induction, we deliberately cast doubt on the effectiveness of this document: the document was incomplete, it was not delivered in a timely manner, it was inaccurate. One doubt that we could not immediately eliminate was the matter of its timely delivery. Investigation found that the regular cycle was not always being maintained.

The correction was to add a metric to the Quarterly Quality Management Review measuring the timely distribution of the document.

Again, the question "is the document regularly sent to customers?" would be answered positively. But by casting doubt on that, we were able to identify, and resolve, a problem.

5.3 Example: Fault Injection Testing

QNX carries out fault injection testing on its operating system kernel. As part of the eliminative induction, doubt was deliberately thrown at this testing: it is not performed often enough, it is not targeted at crucial data structures, it does it reflect real random errors that occur in the field, etc.

It was found that it was not possible completely to eliminate the doubt about the correspondence of the testing method with real random errors: the testing was not truly representative of those errors.

Steps have therefore been taken to improve the verisimilitude of the testing.

6. Analysis

The obvious question is why these problems were not discovered during previous Assurance Case preparations.

QNX found that this was not due to any intent to mislead, but:

- asking engineers, *"How could this evidence be invalid or inadequate?"* is a really fertile way of using confirmation bias positively. Many possible problems were raised and it should be said that most were eliminated.
- changing the question from, *"Is process X being followed?"* to *"Can you think of any example of process X not being completely followed?"* gave the engineers the opportunity to bring dormant concerns into the open: *"Well yes, the process is generally followed, but when situation Y occurs, it is sometimes inadvertently skipped. I've been a bit worried about that for some time, but it doesn't happen often and I've never had the opportunity to mention it before"*.

  A similar inversion of the question for technical matters also led to the discovery of design and implementation (rather than process) problems.

7. Summary

With the new approach, QNX found that it was not necessary to prepare two Assurance Cases: one using GSN to satisfy an auditor and one using BBNs to satisfy internal Safety Engineers. This represented a significant reduction in work, while producing a better Assurance Case and a safer product.

QNX followed this work up with discussions with two of the authors of reference [2]: John Goodenough and Chuck Weinstock. This discussion is ongoing.

Exploration still continues into the practicality of quantifying the notation in two areas:

1. What level of confidence do we place in this piece of evidence?
2. what relative weights do we put on these two sub-arguments (allowing a sensitivity study to be made)?

References

[1] J. GOODENOUGH, C. WEINSTOCK, AND A. KLEIN, *Eliminative Argumentation: A Basis for Arguing Confidence in System Properties*, Tech. Rep. CMU/SEI-2015-TR-005, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 2015.

[2] J. B. GOODENOUGH, C. B. WEINSTOCK, AND A. Z. KLEIN, *Eliminative Induction: A Basis for Arguing System Confidence*, in New Ideas and Emerging Results Workshop, International Conference on Software Engineering, 2013.

[3] GSN COMMITTEE, *GSN Community Standard*, 2011.

[4] B. GUO, *Knowledge Representation and Uncertainty Management: Applying Bayesian Belief Networks to a Safety Assessment Expert System*, in Proceedings. 2003 International Conference on Natural Language Processing and Knowledge Engineering, 2003.

[5] C. HADDON-CAVE, *The NIMROD Review*, tech. rep., Her Majesty's Stationery Office, 2009.

[6] C. HOBBS, *Confirmation Bias within Safety Case Arguments*, in 2016 Safety Critical Systems Symposium, SSS '16, Brighton, UK, 2016, Safety-Critical Systems Club.

[7] C. Hobbs and M. Lloyd, *The application of bayesian belief networks to assurance case preparation*, in 2012 Safety Critical Systems Symposium, SSS '12, Bristol, UK, 2012, Safety-Critical Systems Club.

[8] International Organisation for Standardisation, *ISO/DIS 26262-10 - Road vehicles - Functional safety - Part 10 Guideline on ISO26262*, tech. rep., International Organisation for Standardisation, Geneva, Switzerland, July 2012.

[9] B. Littlewood, L. Strigini, D. Wright, and P.-J. Courtois, *Examination of Bayesian Belief Network for Safety Assessment of Nuclear Computer-based Systems*, DeVa TR No, 70 (1998).