Making the Implicit Explicit: Towards An Assurance Case for DO-178C

C. Michael Holloway; NASA Langley Research Center; Hampton, Virginia, USA

Keywords: assurance case, translation, aviation, correctness, standards

Abstract

For about two decades, compliance with Software Considerations in Airborne Systems and Equipment Certification (DO-178B) has been the primary means for receiving regulatory approval for using software on commercial airplanes. A new edition of the standard, DO-178C, was published in December 2011, and regulatory bodies have started the process towards recognizing this edition. The stated purpose of DO-178C remains unchanged from its predecessor: providing guidance "for the production of software for airborne systems and equipment that performs its intended function with a level of confidence in safety that complies with airworthiness requirements." Within the text of the guidance, little or no rationale is given for how a particular objective or collection of objectives contributes to achieving this purpose. Thus the assurance case for the document is implicit. This paper discusses a current effort to make the implicit explicit. In particular, the paper describes the current status of the research seeking to identify the specific arguments contained in, or implied by, the DO-178C guidance that implicitly justify the assumption that the document meets its stated purpose.

Introduction

For about two decades, compliance with Software Considerations in Airborne Systems and Equipment Certification (DO-178B) (ref. 1) has been the primary means for receiving regulatory approval for using software on commercial airplanes. Despite frequent and occasionally strident criticisms of the standard from various quarters, the empirical evidence is quite strong that it has been successful. Not only has no fatal commercial aircraft accident been attributed to a software error, many of the technological improvements that have been credited with significantly reducing the accident rate have relied heavily on software. For example, controlled flight into terrain—once one of the most common accident categories—has been nearly eliminated by Enhanced Ground Proximity Warning Systems, which are software-intensive (ref. 2).

A new edition of the standard, DO-178C, was published by the issuing bodies in late 2011 (ref. 3). New editions of two associated documents were also published at the same time: DO-278A—Software Integrity Assurance Considerations for Communication, Navigation, Surveillance and Air Traffic Management (CNS/ATM) Systems (ref. 4), and DO-248C—Supporting Information for DO-178C and DO-278A (ref. 5). Additionally four new guidance documents were published simultaneously to address specific issues and techniques: DO-330—Software Tool Qualification Considerations (ref. 6); DO-331—Model-Based Development and Verification Supplement to DO-178C and DO-278A (ref. 7); DO-332—Object-Oriented Technology and Related Techniques Supplement to DO-178C and DO-278A (ref. 8); and DO-333—Formal Methods Supplement to DO-178C and DO-278A (ref. 9). These seven documents have not yet received official regulatory authority approval at the time of this writing, but the regulatory bodies are well along in the process towards recognizing them[1].

The stated purpose of DO-178C remains essentially unchanged from its predecessor: providing guidance "for the production of software for airborne systems and equipment that performs its intended function with a level of confidence in safety that complies with airworthiness requirements." In DO-178B little or no rationale is given for how a particular objective or collection of objectives contributes to achieving this purpose. Thus, the assurance case for the document is implicit. Although empirical evidence suggests that this implicit assurance case has been adequate so far, its implicitness makes determining the reasons for this adequacy quite difficult. Without knowing the reasons for past success, accurately predicting whether this success will continue into the future is problematic.

DO-178C is also mostly rationale-free, but the revised edition of DO-248C includes a new section: 'Rationale for DO-178C / DO-278A'. This rationale section provides a basis from which building an explicit assurance case may

[1] The European Organisation for Civil Aviation Equipment (EUROCAE) uses a different document numbering scheme, but the content of the documents is otherwise identical. For example, DO-178C is identical to ED-12C. For simplicity, only the DO-numbering is referenced in this paper.

be feasible. An effort to build such a case began in September 2012, under the joint sponsorship of the Federal Aviation Administration (FAA) and the National Aeronautics and Space Administration (NASA)[2]. Preliminary work was described in (ref. 10). This paper describes the current status of the research. The remainder of the paper is organized as follows. Section 2 provides background material. Section 3 describes the process followed so far in uncovering the implicit assurance case. Section 4 provides excerpts from the current draft assurance case. Section 5 presents concluding remarks.

Background

Fully understanding this paper requires at least a passing familiarity with DO-178B/C, the assurance case concept, and the Goal Structuring Notation (GSN) for expressing assurance cases. This section provides background information on these subjects for readers who do not already possess the requisite knowledge.

About DO-178C: Appendix A in DO-178C (ref. 3) contains a summary of the history of the DO-178 series of documents. The information in this section is derived from the appendix. The initial document in the series was published in 1982, with revision A following in 1985. Work on revision B began in the fall of 1989; the completed document, which was a complete rewrite of the guidance from revision A, was published in December 1992. Among many other changes, the B version introduced the notion of five different possible software levels, with Level A denoting the highest level (for which satisfying the most rigorous objectives was required), and Level E denoting the lowest level (for which satisfying no objectives was required). The B version also introduced annex tables to summarize the required objectives by software level.

Twelve years after the adoption of DO-178B, RTCA[3] and EUROCAE moved to update the document by approving the creation of a joint special committee / working group in December 2004 (SC-205/WG-71). This group started meeting in March 2005, and completed its work in November 2011. The terms of reference for the group called for (among other things) maintaining an "objective-based approach for software assurance" and the "technology independent nature" of the objectives. The special committee/working group was also directed to seek to maintain "backward compatibility with DO-178B" except where doing so would fail to "adequately address the current states of the art and practice in software development in support of system safety", "to address emerging trends", or "to allow change with technology." The seven documents produced by the efforts—three updates and four entirely new—were enumerated in the introduction.

As a result of the terms of reference and operating instructions under which it was produced, DO-178C is an update to, as opposed to a re-write or substantial revision of, DO-178B. Differences between the B and C versions include corrections of known errors and inconsistencies, changes in wording intended for clarification and consistency, an added emphasis on the importance of the full body of the document, a change in qualification criteria for tools and the related creation of a separate document for tool qualification, modification of the discussion of system aspects related to software development, closing of some perceived gaps in guidance, and the creation of technology-specific supplements for formal methods, object-oriented technology, and model-based design and verification.

About assurance cases: The concept of an assurance case is a generalization of the safety case concept. A safety case is "a structured argument, supported by a body of evidence that provides a compelling, comprehensible and valid case that a system is safe for a given application in a given operating environment" (ref. 11). Safety is the paramount attribute. Claims are made concerning the achievement of an acceptable level of safety, and the arguments and evidence are focused on providing justified confidence that those claims are satisfied. An assurance case, on the other hand, is concerned about providing justified confidence that claims are satisfied about additional desired attributes such as functionality, performance, or security.

Claims, arguments, evidence, context, and assumptions constitute five necessary components of a good assurance case (ref 12). *Claims* are statements about desired attributes. Other names that are used for the same concept include

[2] The joint sponsorship is under the auspices of Interagency Agreement IAI-1073: *Verification and Validation for Complex Systems*. Although the FAA has partially funded the work described in this paper, it played no part in the approval process for this paper.

[3] Once upon a time, RTCA was an abbreviation for Radio Technical Commission for Aeronautics; since 1991 the four letters have been the freestanding name of the organization.

goals, propositions, and conclusions. In a full assurance case, there will likely be many claims that must be shown to hold, at varying levels of generality. An example of a high-level claim is *The system is sufficiently safe to satisfy airworthiness requirements within its intended environment*. Examples of claims with an increasing level of specificity are as follows: *Credible hazards have been identified*; *Hazard H has been eliminated by design*; and *Hardware component M has an acceptably long expected mean-time-to-failure*.

Arguments show how the stated claims are supported by, or justifiably inferred from, the available evidence. Other terms sometimes used for the same concept include strategies, warrants (ref. 13), and reasons. *Evidence* refers to the available body of known facts related to system properties or the system development processes. Data, facts, and solutions are synonymous terms. Examples of evidence include hazard logs, testing results, properties of materials, and mathematical theorems.

*Contex*t refers to any information that is needed to provide definitions or descriptions of terms, or to constrain the applicability of the assurance case to a particular environment or set of conditions. As example, the context for the claim The software performs its intended function with a level of confidence in safety that complies with airworthiness requirements would likely include the applicable airworthiness requirements, a description of the intended function of the software, and any constraints on the environment in which the software is expected to be used. *Assumptions* are statements on which the claims and arguments rely, but which are not elaborated or shown to be true in the assurance case. As an example, an argument concerning safety that shows all identified hazards have been eliminated relies on the assumption All credible hazards have been identified.

Each of these components is present implicitly in the collective minds of the developers of any successful engineered system. An assurance case simply provides a means for ensuring that all of this implicit knowledge is documented explicitly in a form that can be examined carefully and critically, not only by the developers, but also by others. An active research community is exploring how to best create, express, analyze, improve, and maintain assurance cases (refs. 14-22).

About GSN: The Goal Structuring Notation is one of the most popular notations for expressing assurance cases (ref. 23). Some of the primary symbols of the notation are illustrated in figure 1. Text within these symbols is used to provide content and a convenient means of referring to individual elements.

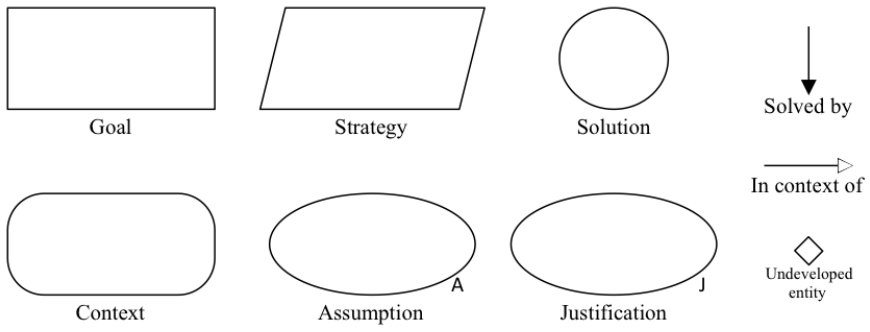

Figure 1: Main elements of GSN

The concepts represented by most of these elements have already been described. A *justification* gives the rationale for why a particular strategy or goal is acceptable. To construct an argument, the elements of the GSN notation are linked together using the *in context of* or *solved by* directed lines. The *undeveloped entity* symbol is appended to the bottom of a goal or strategy to indicate that the particular line of argument requires further development. GSN will be used in the rest of this paper to express selected portions of the preliminary assurance case that has been developed in the research. The next section explains the process that has been followed so far.

The Process

The work to date has included the following five primary activities:

- Careful and continuing study of DO-178C and DO-248C
- Review of previous work
- Assessment of applicability of confidence argument concept
- Preliminary classification of objectives
- Selection of an approach for creating arguments

The rest of this section describes each of these activities in appropriate detail.

Study: Careful study of the text of DO-178C and relevant sections of DO-248C was the first activity in the project, and will continue throughout it. The initial study was focused on finding important context and assumptions on which the guidance rests. The results of the study were described fully in (ref. 10). For the purposes of this current paper, one item of context and one fundamental assumption are worth repeating.

The stated purpose of DO-178C, which was quoted in the introduction section of this paper, identifies a critical item of context: the airworthiness requirements. These airworthiness requirements are defined outside of DO-178C in the Code of Federal Regulations Title 14 (ref. 24). Different parts of Title 14 apply depending on the category of vehicle. Applicable categories include transport category airplanes (part 25); normal, utility, acrobatic, and commuter airplanes (part 23); transport category rotorcraft (part 29); normal category rotorcraft (part 27); products and parts (part 21); and engines (part 33). These differences mean that the top-level context for an explicit assurance case for software for a transport category airplane will be different from the context for a commuter category airplane. The former will refer to part 25, while the latter will refer to part 23.

Whereas the airworthiness context is clearly stated in DO-178C, a fundamental assumption of the guidance is discernable only through inferences from the text. This assumption involves the relationship between safety and correctness. Although in the general case, these two concepts are not equivalent (ref. 12), DO-178C rests on the assumption that within the constraints established by the guidance, establishing justifiable confidence in the correctness of the software is sufficient to establish justifiable confidence that the software does not contribute to unsafe conditions. The constraints underlying this assumption include the adequacy of the system safety processes conducted outside of the scope of DO-178C (refs. 25, 26), the effective allocation of requirements (including the requirements needed to ensure safety) to software, and the analysis by system safety processes of any new requirements that arise during software development[4].

Review: Another activity undertaken was the review of previous, related research. No published work was found that attempted to accomplish identical goals to the current effort, but two projects were uncovered that dealt with related aspects of assurance cases and DO-178B.

The MITRE Corporation conducted an effort to map three different standards into an assurance case framework (ref. 27). The primary purpose of this effort was to explore two primary hypotheses: all assurance cases have similar components, and an assurance standard implies the structure. One of the three standards used in the study was DO-178B. The created assurance case was structured rigidly around the DO-178B chapters. The top- level claim was *DO-178B Software Considerations are taken into account*. Sub-claims were given for each of the DO-178B chapters 2 – 9. For example, sub-goals included the following: *2.0 System Aspects are taken into account*, *4.0 Software Planning Process is executed*, *5.0 Software Development Process is executed as planned*, and *9.0 Certification Liaison process is properly established & executed*.

As best as can be determined from the published material, the effort concentrated on translating the textual and tabular form of DO-178B into a graphical form with as little interpretation or abstraction as possible. This differs substantially from the current research, which is concentrating on discovering the underlying implicit assurance case, not rigidly translating one form of concrete expression into another form.

[4] In DO-178C (and B) terminology, such requirements are called *derived requirements*. Derived requirements must be passed back to system processes, including system safety processes, for analysis of (among other things) potential safety implications.

Researchers at the University of York and QinetiQ in the United Kingdom conducted the other related previous work (ref 28). The primary goal of this research was to explore ways to justify substitution of one technology for another. In particular, the emphasis was to develop arguments showing that the evidence produced by replacements for testing (such as formal proof) could be at least as convincing as the evidence produced by testing. As part of this research, certain aspects of the testing-related objectives of DO-178B were explored and GSN representations were produced. Unpublished results from the research were submitted to SC-205/WG-71, and considered by the sub-group responsible for creating the document that eventually become DO-333. These results have been helpful in considering various approaches to discovering and expressing a full assurance case for DO-178C.

Assess: Recent research from the University of York and the University of Virginia (ref. 18) has been even more helpful towards that end. This research introduces the idea of a *confidence argument* to accompany a primary safety argument. The safety argument documents the arguments and evidence related to direct claims of safety; the confidence argument documents the arguments and evidence related to the sufficiency of confidence in the primary argument. This separation into two different argument structures differs from the prevailing practice of intermixing concerns of safety and confidence in a single unified argument, and offers promise of eliminating or mitigating some of the difficulties recognized in the prevailing approach (ref. 29). Although the paper is presented in terms of a safety case, the authors acknowledge that the general concept applies equally to any property of interest.

Assessing whether the concept is appropriate for expressing the assurance argument for DO-178C was the third major activity undertaken to date. The answer was a definite yes. Even a cursory reading of the guidance reveals that it contains a mixture of objectives about the desired properties of the final software product, intermediate products, and the processes used to develop the product. A more careful reading shows that some of these objectives are naturally part of a primary argument about correctness of the final software, some are naturally part of a confidence argument that justifies appropriate belief in the sufficiency of the correctness argument, and some are a bit difficult to classify.

Classify: Conducting a preliminary classification of DO-178C objectives thus became the fourth major activity in the project. The first attempt at classification was based on the notion that every objective would likely correspond to a claim in either a correctness or confidence argument. It did not take very long to realize that this notion was too simplistic. The range of possibilities for logical correspondence of objectives not only includes claims, but also evidence, context, assumptions, and justifications. Based on this realization, the first classification was abandoned, and a second attempt was completed using the following three categories:

{1} The objective is likely to appear in some form as a claim or evidence in the primary argument.
{2} The objective is likely to appear in some form as a claim or evidence in a confidence argument.
{3} The objective is likely to appear as context, assumption, or justification in an argument (rather than as a claim or evidence).

Three examples of objectives placed in to category {1} are the following: *High-level requirements comply with system requirements* (this objective is summarized in row 1 of Table A-3, and thus often referred to as A-3.1); *Executable Object Code complies with high-level requirements* (A-6.1); and *Executable Object Code complies with high-level requirements* (A-6.5). Each of these objectives concerns properties of the final software product, and thus is directly related to a primary assurance argument. If one of these objectives is not satisfied, then it is not possible for the software to satisfy goals concerning its correctness.

Category {2} objectives include, for example, *Software plans comply with this document* (A-1.6); *Test coverage of software structure (statement coverage) is achieved* (A-7.7); and Problem reporting, change control, change review, and configuration status accounting are established (A-8.3). The reasons for the classification of these three differ. Objectives A-1.6 and A-8.3 refer to a property of the process not the product, and thus properly relate to confidence. Objective A-7-7 concerns a property of the product, but the objective does not necessarily have to be satisfied in order for the final software to satisfy goals about its correctness. It is possible for the running software to correctly implement its requirements even if the testing of the software did not cover all statements; however, higher confidence in the correctness of the software is justified if the objective is satisfied than if it is not.

Finally, the following two objectives are examples of category {3}: *The activities of the software life cycle processes are defined* (A-1.1); and *The means of compliance is proposed and agreement with the Plan for Software Aspects of Certification is obtained* (A-10.2). These objectives set part of the context within which the primary and confidence arguments reside, but are not appropriate as either claims or evidence in those arguments.

The full initial classification is summarized as follows:

- Of the 71 Level A objectives, 21 were determined to be likely to appear in some form in a primary argument, 36 in a confidence argument, and 14 as context.
- For Level B's 69 objectives, the breakdown was 20 primary, 35 confidence, and 14 context.
- For Level C's 63 objectives, the breakdown was 18 primary, 31 confidence, and 14 context.
- For Level D's 26 objectives, the breakdown was 8 primary, 10 confidence, and 8 context.

Whether these results will remain consistent throughout the remainder of the project is an open question. It seems likely that some changes will result as the primary and confidence arguments are developed and reviewed. Potential changes include not only reclassification from one category to another, but also combining multiple objectives into a single entity within an argument, and removal of some objectives from the argument entirely.

Select: Based on the results of the four activities already described, the fifth activity undertaken was determining how best to proceed in creating the initial candidate arguments. Three main questions were considered in making this determination.

Question 1 was, "What software level should be considered first?" In favor of starting with level A is the fact that the higher the level, the more important the assurance case is; thus, articulating an explicit assurance case for level A has more value than for lower levels. In favor of starting with level D is the fact that its relatively small number of objectives simplifies the tasks of discovering and articulating the explicit case, and makes reviewing the case by others easier. By increasing the likelihood of receiving constructive feedback on the initial effort, starting with level D seems likely to provide the best chance that the final product will be of high quality. So, the answer to the question was determined to be "Level D."

The second question considered was, "What notation will be used?" No single notation is ideal for everyone who may be interested in the results of the work (ref. 22); however, insufficient resources are currently available to allow expression in multiple notations. As has been already noted earlier in the paper, the answer to this question was determined to be "GSN."

"Will the developed assurance cases necessarily adhere to the DO-178C chapter / table format?" was the third main question considered. Adherence to such a format has characterized the previously published work, and dominated initial thinking in this project. Structuring sub-goals to correspond to the Annex A objectives tables seemed a natural way to proceed at first. Further reflection, however, suggested that such a structure would have two significant disadvantages: it would tend to emphasize the tables at the expense of the full text (avoiding such an emphasis was one of the goals of the C revision), and it would likely overly constrain the expression of the arguments. So, the answer to this question was determined to be "No."

With the answers determined to these questions, the initial articulation of a primary assurance argument and associated confidence arguments for Level D software could begin. The current results from that effort are presented in the next section.

A Partial Case

The current draft primary assurance argument is presented first, divided into three parts. Afterwards a portion of one confidence argument is presented.

Primary argument: The top-level of the primary assurance argument created so far is shown in Figure 2. The overall goal *Software performs intended function at acceptable level of safety for level D* is derived from the stated purpose of DO-178C, modified for the software level. Three items of context are identified as necessary for this goal to

make sense: a description of the software's intended function, the definition of software level D, and the relevant parts of the airworthiness requirements that define what constitutes an acceptable level of safety. Only the definition of software level D is provided directly in DO-178C; the others are external to the document. A critical assumption on which the entire argument rests is that the assignment of level D to the software is correct.

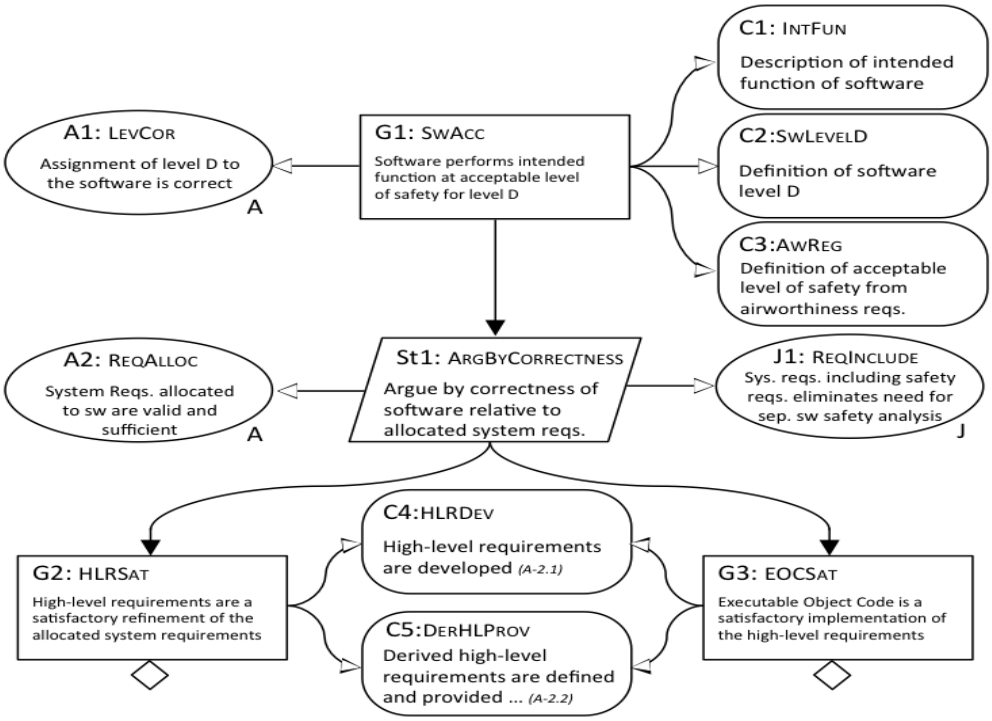

Figure 2: Beginning of primary argument for level D software

The DO-178C objectives and activities for Level D software imply that the implicit argument for the top-level goal G1 is based on showing that the software is correct relative to the allocated system requirements. This implicit argument relies for its cogency on (a) the assumption that the allocated system requirements are valid and sufficient with respect to the software's intended function; and (b) the justification discussed in the previous section explaining the relationship between correctness and safety in the presence of valid and sufficient requirements.

For Level D software, arguing by correctness involves two sub-goals: G2:HLRSAT and G3:EOCSAT. The former involves showing an appropriate relationship between the developed high-level requirements and the system requirements; the latter involves showing that the developed executable object code implements the high-level requirements. Both of these goals have meaning only within the context of high-level requirements being developed (which is DO-178C objective A-2.1), and any derived high-level requirements being provided to the system processes, including the system safety assessment processes (A-2.2). Figures 3 and 4 show further refinements of G2 and G3 respectively.

Demonstrating satisfaction of G2 comprises three sub-goals, which correspond directly to the three DO-178C objectives related to the verification of outputs of software requirements process (summarized in Table A-3) that are imposed for level D software: showing that the high-level requirements comply with system requirements (A-3.1), are accurate and consistent (A-3.2), and are traceable to system requirements (A-3.6). In the figure, context is shown only for the definition of traceable, so as to simply presentation for this paper; but the final complete assurance case will need to include context for definitions / descriptions of comply, accurate, and consistent. According to DO-178C, the evidence for satisfaction of these three objectives is contained in the Software Verification Results, which is a data item described in Chapter 11 of the guidance.

The argument for satisfaction of G3 refines into four sub-goals. Three of these sub-goals correspond to the three level D applicable objectives for testing of outputs of integration process (summarized in Table A-6); the remaining sub-goal corresponds to the only applicable objective for verification of outputs of software design process (Table A-4). All four applicable software development process (Table A-2) objectives constitute part of the relevant context for this part of the argument. One of these objectives is divided into two parts, because the portion of the objective dealing with the production of Executable Object Code (EOC) seems appropriately part of the context for G3, while the part dealing with loading of the code onto the target computer seems to be better attached to the goal concerning compatibility of EOC and target computer. The evidence for the achievement of the four sub-goals is taken from the three specific data items shown. As was the case for the G2 refinement, the refinement here for G3 shows only some of the contextual items that will need to be included in a final, complete assurance case.

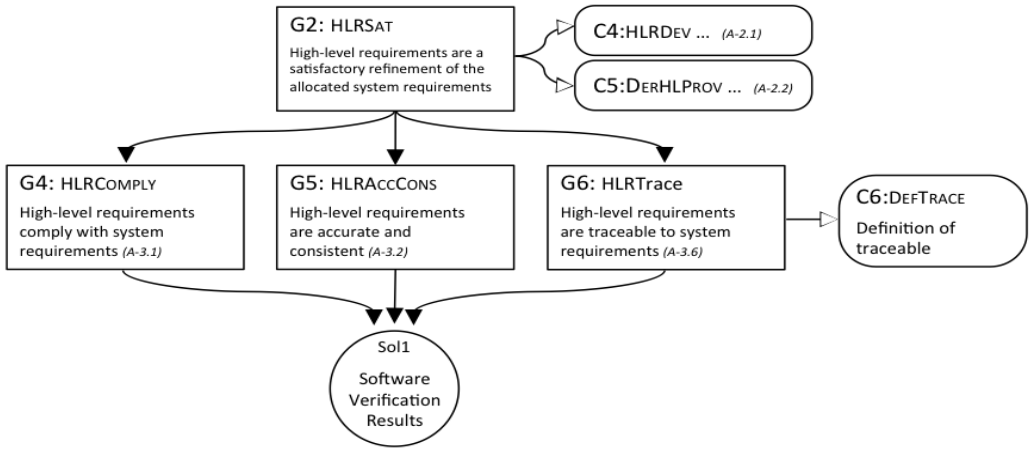

Figure 3: Refinement of G2:HLRSAT

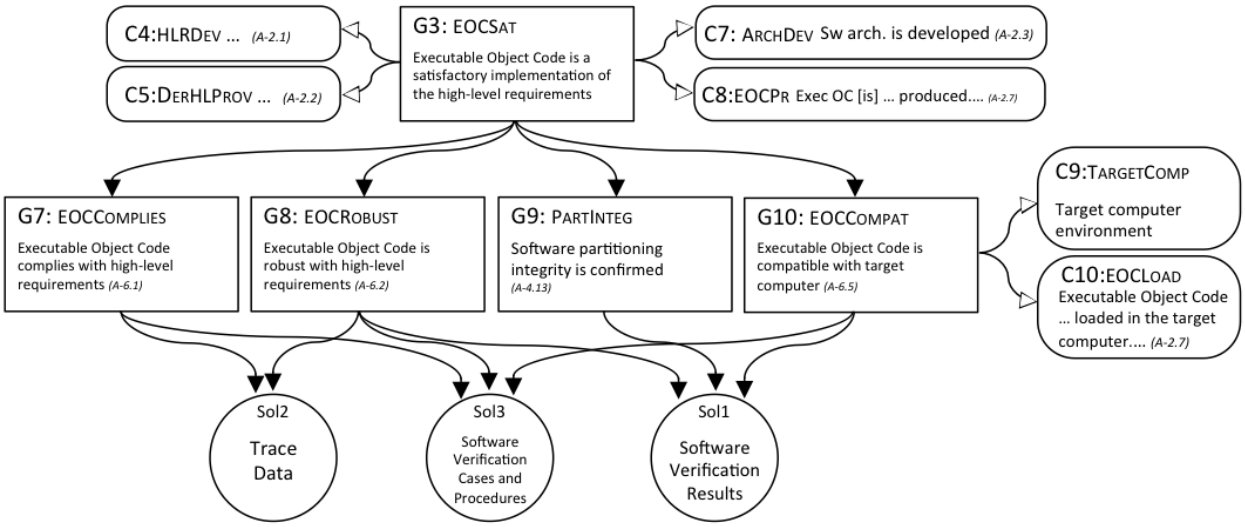

Figure 4: Refinement of G3:EOCSAT

With the exception of the necessary additional contextual items already mentioned, and the absence of any goals or evidence concerning the objective related to Parameter Data Item files (A-5.8), figures 2-4 represent a complete draft articulation of the implicit primary assurance argument implied by DO-178C for level D software.

Confidence argument: Figure 5 represents a portion of one of the associated confidence arguments that have been developed so far. The goal of this argument is to illustrate the implicit reasoning in DO-178C that justifies the belief that the data items required are adequate evidence for the satisfaction of the low-level goals in the primary argument. The confidence argument relies on the adequacy of the testing of the software and of the configuration management

processes in place. For level D software, the only required measure of testing adequacy is that coverage of the high-level requirements has been achieved (A-7.3); thus the testing branch of the argument has only one sub-goal.

On the other hand, DO-178C imposes on level D software the same six objectives as are required for higher software levels for configuration management; so there are six sub-goals on this branch. Three of these are shown explicitly in the figure, along with the data items that constitute the evidence for their satisfaction.

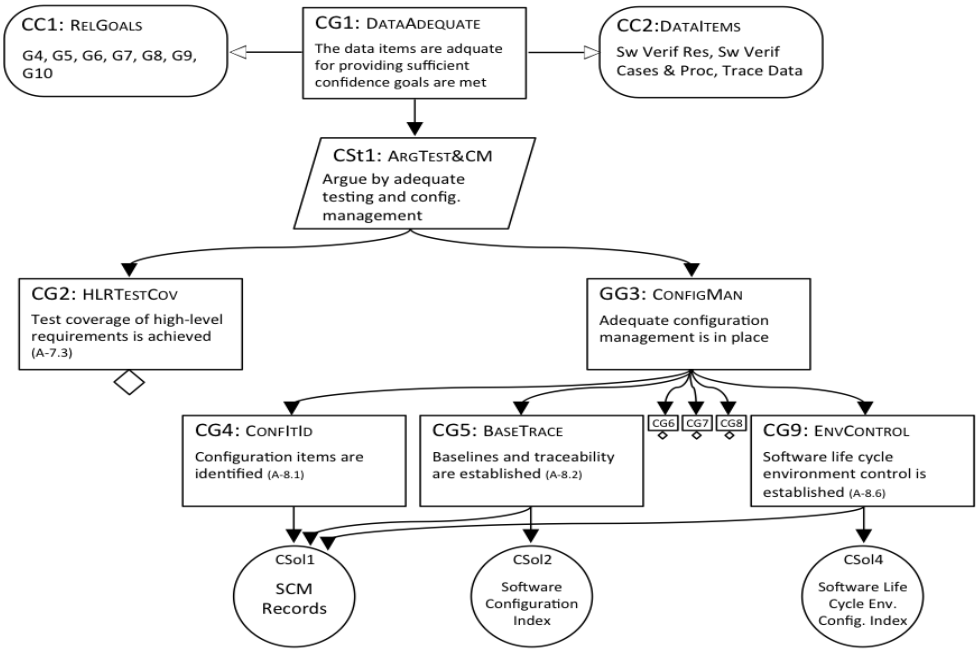

Figure 5: A partial confidence argument

Confidence arguments are also associated with (a) the reasoning from G1 through St1 to the refinement into G2 and G3; (b) the refinement of G2; and the refinement of G3. Barring unforeseen events, these confidence arguments will have been constructed by the time this paper is presented at the conference.

Concluding remarks

This paper explained the current status of on-going research seeking to uncover the implicit assurance case that resides within the DO-178C guidance. Relevant background material was presented, completed activities were enumerated, and excerpts from the current draft articulation of an assurance case for level D software were illustrated and discussed. Comments on the draft case are welcome.

The current schedule for the research calls for completing the discovery and articulation of assurance cases for the full DO-178C guidance before the end of 2013. Assessing the utility and feasibility of conducting a similar activity for one or more of the technology supplements will begin shortly thereafter.

Three potential benefits are anticipated from successful completion of the research. First, making explicit the reasoning on which the guidance rests should enable effective analysis of the adequacy of the reasoning. Such an analysis would provide a solid foundation on which future modifications to the guidance could be based. Second, the existence of an explicit assurance case for the guidance should facilitate intelligent conversations about the relative efficacy of DO-178C and other existing or proposed approaches for demonstrating compliance with airworthiness regulations. The third potential benefit is a little bit more nebulous than the other two, but perhaps more important for the system safety community worldwide than either of them. Rightly or wrongly, DO-178 is thought by many to be a prototypical example of a prescriptive standard (refs. 12, 14). Expressing its essence as an assurance case may improve cooperation and mutual understanding between supporters of prescriptive-style standards and supporters of goal-based-style standards. At least one may hope.

References

1. RTCA. “Software Considerations in Airborne Systems and Equipment Certification.” DO-178B. 1992.

2. Rushby, J. “New Challenges in Certification of Aircraft Software.” *Proceedings of the 11th International Conference on Embedded Software (EMSOFT)*. Taipei, Taiwan, 2011

3. RTCA. “Software Considerations in Airborne Systems and Equipment Certification.” DO-178C. 2011.

4. RTCA. “Software Integrity Assurance Considerations for Communication, Navigation, Surveillance, and Air Traffic Management (CNS/ATM) Systems.” DO-278A. 2011.

5. RTCA. “Supporting Information for DO-178C and DO-178A.” DO-248C. 2011.

6. RTCA. “Software Tool Qualification Considerations.” DO-330. 2011.

7. RTCA. “Model-Based Development and Verification Supplement to DO-178C and DO-178A.” DO-331. 2011.

8. RTCA. “Object-Oriented Technology and Related Techniques Supplement to DO-178C and DO-178A.” DO-332. 2011.

9. RTCA. “Formal Methods Supplement to DO-178C and DO-178A.” DO-333. 2011.

10. Holloway, C.M. “Towards Understanding the DO-178C / ED-12C Assurance Case.” *7th IET International Conference on System Safety, Incorporating the Cyber Security Conference.* Edinburgh, 2012.

11. UK Ministry of Defence. *Defence Standard 00-56 Issue 4: Safety Management Requirements for Defence Systems*. 2007.

12. Knight, J. *Fundamentals of Dependable Computing for Software Engineers*. Boca Raton, Florida: CRC Press, 2012.

13. Toulmin, S. E. *The Uses of Argument, Updated Edition*. Cambridge University Press, 2003.

14. Hawkins, R.; Habli, I.; Kelly, T.; McDermid, J. “Assurance cases and prescriptive software safety certification: A comparative study.” *Safety Science*. Vol 59, 2013.

15. Goodenough, J. B.; Weinstock, C. B.; Klein, A. Z. “Toward a Theory of Assurance Case Confidence.” CMU-SEI-2002-TR-002, September 2012.

16. Graydon, P; Habli, I; Hawkins, R.; Kelly, T; Knight. J. “Arguing Conformance.” *IEEE Software* 29 (3), 2012.

17. Denney, Ewen; Pai, Ganesh. “A Lightweight Methodology for Safety Case Assembly.” *Proceedings of the 31st International Conference on Computer Safety, Reliability and Security (SafeComp '12)*, Magdeburg, Germany, 2012.

18. Hawkins, R; Kelly, T; Knight, J.; Graydon, P. “A New Approach to Creating Clear Safety Arguments.” *Advances in Systems Safety*. C. Dale and T. Anderson (eds). Springer-Verlag, 2011.

19. Yuan, T.; Kelly, T. “Argument Schemes in Computer System Safety Engineering.” *Informal Logic* 31 (2), 2011.

20. Bloomfield, R.; Bishop, P. “Safety and Assurance Cases: Past, Present and Possible Future.” *Making Systems Safer*. C. Dale and T. Anderson (eds). Springer-Verlag, 2010.

21. Hawkins, R.; Kelly, T. "A Systematic Approach for Developing Software Safety Arguments." *Proceedings of the 27th International System Safety Conference*. Huntsville, Alabama, 2009.

22. Holloway, C. M. "Safety Case Notations: Alternatives for the Non-Graphically Inclined?" *Proceedings of the 3rd IET International System Safety Conference*. Birmingham, UK, 2008.

23. GSN Committee. Draft GSN Standard Version 1.0. <http://www.goalstructuringnotation.info/> (last accessed June 3, 2013).

24. Federal Aviation Administration. "Standard Airworthiness Certification: Regulations – Title 14 Code of Federal Regulations." <http://www.faa.gov/aircraft/air_cert/airworthiness_certification/std_awcert/std_awcert_regs/regs/> (last accessed June 12, 2013).

25. Society of Automotive Engineers. "Guidelines for Development of Civil Aircraft and Systems." SAE ARP 4754a, 2010.

26. Society of Automotive Engineers. "Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment." SAE ARP 4761, 1996.

27. Ankrum, T. Scott; Kromholz, Alfred H. "Structured Assurance Cases: Three Common Standards." Proceedings of the Ninth IEEE International Symposium on High-Assurance Systems Engineering (HASE'05). Heidelberg, Germany, 2005.

28. Galloway, A; Paige, R. F.; Tudor, N. J.; Weaver, R. A.; McDermid, J. "Proof vs. Testing in the Context of Safety Standards." *The 24th Digital Avionics Systems Conference (DASC)*, Washington D.C., 2005.

29. Haddon-Cave, C. *The Nimrod Review*. London: The Stationary Office, 2009. <http://www.official-documents.gov.uk/document/hc0809/hc10/1025/1025.pdf> (last accessed June 12, 2013).

Biography

C. Michael Holloway, Senior Research Engineer, NASA Langley Research Center, 100 NASA Road, Hampton VA 23681-2199, telephone – (757) 864-1701, facsimile – (757) 864-4234, e-mail – c.m.holloway@nasa.gov.

C. Michael Holloway is a senior research computer engineer at NASA Langley Research Center. His primary professional interests are system safety and accident analysis for software-intensive systems. He is a member of the IEEE, the IEEE Computer Society, and the International System Safety Society.