

An Assessment Basis for Aeronautical System Assurance Cases

Ewen Denney and Ganesh Pai
KBR / NASA Ames Research Center
Moffett Field, California, USA
{ewen.denney, ganesh.pai}@nasa.gov

Abstract—Assurance cases offer certification applicants the latitude of choosing development processes suitable to their constraints whilst relying upon explicit arguments to provide justified confidence that the safety intent of the applicable regulations will be met through alternative means of compliance. However, it is possible to produce superficially plausible assurance arguments asserting that the relevant safety objectives have been met, due to which a clear approach is needed to discriminate between arguments that are fit for purpose and those that are inadequate. Moreover, a well-defined basis for the extent, strength, and proportionality of evidence is necessary for sufficient assurance. This paper presents an assurance case assessment basis: a collection of structured documents through which a certification applicant and regulator can reach consensus and explicitly document the foundations and principles by which a series of assurance cases will be planned, developed, reviewed, and the criteria upon which they will ultimately be accepted. An example of a high altitude long endurance unmanned aircraft system is used to illustrate the related concepts.

Index Terms—Assessment, Assurance Cases, Approval, Certification, Machine Learning, Safety Cases

I. INTRODUCTION

The prevailing processes for aircraft certification and approval¹ involve an iterative, document-driven dialog between the applicant and the regulatory authority [1]. Regulatory review of lifecycle data produced by a certification applicant leads to actions that are meant to resolve findings, observations, and issues. Ultimately, the outcome is a state of shared and justified confidence in the safety of the system and the functionality being approved. With the aim of streamlining certification, and—more recently—for approving the deployment of new and novel technologies such as machine learning (ML) into aircraft systems, the US civil aviation regulator, i.e., the Federal Aviation Administration (FAA), has been investigating the use of assurance cases.

The research in this paper was funded by the Federal Aviation Administration (FAA). A part of the work reported herein was performed under Contract No. 80ARC020D0010 with the National Aeronautics and Space Administration (NASA). This paper is disseminated under the sponsorship of the U.S. Department of Transportation in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof. The U.S. Government does not endorse products or manufacturers. Trade or manufacturers' names appear herein solely because they are considered essential to the objective of this paper. The findings and conclusions are those of the authors and do not necessarily represent the views of the funding agencies. This document does not constitute FAA policy.

¹Only aircraft, engines, and propellers undergo *certification*, whereas software functionality is *approved*.

Assurance cases (Section II) allow certification applicants the flexibility to choose (non-standard) development processes and analysis methods suitable to their constraints and the criticality of their application as *alternative means of compliance* (AMOC) to aviation regulations. This approach relies upon explicitly stated rationale, e.g., structured arguments, that relate heterogeneous evidence to concretely stated objectives to provide justified confidence that the safety intent of the applicable regulations will be met through an AMOC. However, due to that flexibility, regulatory assessment of an assurance case can be both more challenging and effort intensive.

Moreover, superficially acceptable but deficient assurance arguments can be developed to assert that the relevant safety objectives have been met for a given level of safety-criticality. Deficiencies can include fallacious reasoning, or supporting evidence that is inadequate (inherently problematic), insufficient (applicable but incomplete), or of questionable relevance (for an example, see Section II-B5 and Fig. 2).

As such, there is a general need for: (i) a clear approach to discriminate between assurance arguments that are suitable and those that are inadequate; and (ii) a well-defined basis for the extent, strength, and proportionality of evidence that is necessary for sufficient assurance. Although there has been prior research on assurance case assessment [2]–[4], the applicability to and compatibility with the context of current aircraft certification/approval processes has yet to be determined. The contributions of this paper are thus aimed at addressing the aforementioned technical needs, whilst being compatible with existing review processes during certification/approval.

Specifically, the primary contribution is a notion of *assurance case assessment basis* (ACAB): a collection of inter-related structured documents to facilitate consensus between a certification applicant and the regulator. As will be clarified subsequently (Section III), an ACAB comprises a *development plan*, a *development basis*, and an *evidence basis* for one or more assurance cases. Together, they explicitly document the foundations and principles with which a series of assurance cases will be developed and reviewed, along with the criteria ultimately necessary for their acceptance.

A secondary contribution is a brief discussion of a proposed assurance case assessment methodology that leverages the ACAB, and is inspired by current certification/approval review processes (Section IV). The underlying approach involves an iterative back-and-forth exchange to systematically and

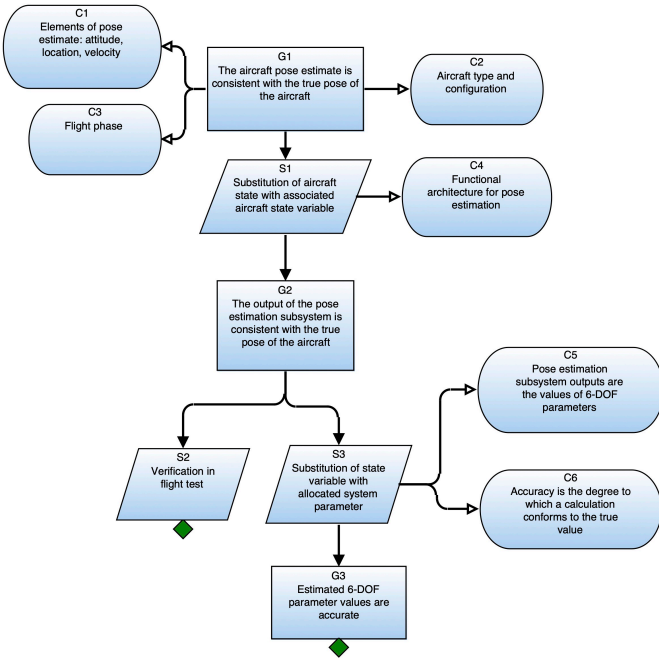


Fig. 1. Example fragment of a graphically depicted argument in GSN.

critically probe, first, the core elements of the ACAB, and then, the assurance cases developed in accordance with the ACAB, to establish that each is acceptable.

We illustrate these contributions (Section V) using an example of a high altitude long endurance unmanned aircraft system, also known as a high altitude platform system (HAPS).

II. BACKGROUND CONCEPTS

A. Assurance Cases

The notion of assurance case considered here is broad, and comprises multiple related artifacts communicating different, related viewpoints on providing justified confidence that a system or service is fit for its intended purpose [5]. For an assurance case whose main focus is safety (known as a *safety case*), core artifacts include at least:

- a *hazard log* and a *requirements log*, that together capture the hazard analyses, risk assessments, resulting safety objectives, and the corresponding risk mitigation requirements;
- a *safety architecture* reflecting the high-level technical approach to risk mitigation;
- *assurance rationale*, captured in the form of structured arguments (Sections II-B1 and II-B3) and argument patterns (Section II-B2); and
- an *evidence log* to record and manage the artifacts that provide confirmation that the safety objectives have been met (Section II-C);

An example of a safety case including excerpts of these assurance artifacts for an automated visual landing (AVL) application integrating machine learning (ML) based functionality is in [6]. The focus of this paper from an assessment standpoint is on the assurance rationale and evidence.

B. Assurance Rationale

1) *Arguments*: A structured assurance argument systematically captures the rationale necessary to comprehensively justify to the stakeholders of a system or service why they can rely upon it. That is, why stakeholders can have sufficient and justified confidence that their needs on system functionality and its dependability attributes (e.g., safety) have been met.

We can represent arguments in a number of ways including: narrative text (using constrained or unconstrained natural language, or domain-specific languages); tables; graphically; or as a combination thereof. Fig. 1 gives an example argument fragment from the assurance case of the AVL system from [6], shown graphically using the Goal Structuring Notation (GSN) [7], created using the assurance case tool Advocate [8]. It explicitly highlights different argument elements: claims and sub-claims (shown as rectangles) are connected by inference rules (shown as parallelograms) each of which are linked by support relationships (shown as filled arrows), along with the relevant context (shown as rounded rectangles connected by hollow arrows). It also indicates when the reasoning elements of an argument need further development (shown by the diamond annotation in Fig. 1).

In general, explicit arguments need not be created for assurance of the functionality and dependability attributes of every system function. Indeed, the current approach to safety assurance of complex aircraft systems and software in the aerospace recommended practices and guidelines [9]–[11] relies on *development assurance*. That entails planned and systematic actions to identify and correct development errors. Although the lifecycle artifacts resulting from development assurance activities serve as evidence for adequate confidence that the safety objectives have been met, the justifying arguments for those activities are usually implicit.

Explicitly stated assurance arguments may be more appropriate when: (i) the rationale for assurance of specific functionality and their related dependability attributes is crucial for the overall understanding of how that functionality contributes to the broader system (or subsystem) context; (ii) the existing development and assurance practices are inadequate; or (iii) there is a need to justify new, non-standard methods and practices to be invoked as an AMOC.

For instance, the argument in Fig. 1 relates the accuracy of ML-based keypoint estimation to the correctness of a pose estimation function. Existing development assurance practices are known to have gaps when applied to ML-based functionality [12]. Thus, the assurance rationale in Fig. 1 aids in understanding why one can conclude with sufficient confidence that ML-based keypoint estimation contributes as required from a system safety standpoint, to the aircraft functions that rely upon correct pose estimation.

In general, an assurance argument may be *inductive* (i.e., its conclusion is probable, but not certain, given its premises) or it may be *deductive* (i.e., the premises logically entail the conclusion). An inductive argument is said to be *strong* when its conclusion is highly likely given its premises, otherwise it

is *weak*. Only strong arguments with true premises are *cogent*, i.e., their conclusions can be accepted. The analogous terms for a deductive argument are *valid* and *sound*.

2) *Argument Patterns*: Reusable structures that capture the commonality of reasoning between different arguments whilst abstracting their details are known as *argument patterns*. For example, a *decomposition* pattern captures the following common reasoning structure: distribute a claim about some concept or property (e.g., correct behavior of a function) into sub-claims (e.g., correct behavior of the implementing items), such that the evidence for those sub-claims and the validity of the composition entails the parent claim. Additional examples of various other argument patterns may be found in [13], [14].

Patterns abstract not only the argument structure, but also the terms and concepts within the argument elements, which may be given as variables and parameters. An argument pattern can be *instantiated* to produce an *instance argument*, i.e., an argument whose structure, variables, and parameters have been replaced with concrete values [14]. Thus, instantiating a pattern in different contexts yield different concrete instance arguments. Additionally, patterns can be *composed* to create compound arguments [15].

3) *Argument Modules*: Modules facilitate separating assurance concerns, providing a mechanism for managing argument organization and size. An argument module groups collections of related arguments and patterns to encapsulate their content through explicit interfaces [7]. Arguments contained within a module are known as *modular arguments* [16].

For example, the correctness argument for some functionality may be separated from the argument pertaining to how that functionality contributes to system safety. Thus, the arguments of a system safety case could be organized such that, say, the safety-related modular arguments are contained within a *safety argument module* that then refers to a *correctness argument module* either entirely, or in part. Indeed, this modular abstraction reflects the high-level structure of the implicit argument for confidence that software contributions to system/aircraft-level failure conditions are acceptable, in the combined application of the accepted guidelines for aircraft system safety [10] and airborne software assurance [11].

4) *Argument Architecture*: An assurance case may contain various modular arguments or compound arguments that result from composing and instantiating different argument patterns. The comprehensibility of such an assurance case may be improved by an *argument architecture* [6], [17], i.e., a general structure conveying a high-level overview of the interrelations of the different arguments.

When an argument architecture is realized as the interconnection of modules (e.g., see Fig. 6), it is analogous to the GSN *module view* [7]. Although, GSN does not have similar views for composed patterns, or compound arguments in general. The notion of argument architecture will be leveraged later (Section III-A) when developing one of the main building blocks of the assurance case assessment basis (ACAB).

5) *Counterarguments*: An argument that challenges or aims to defeat another argument (or its parts) is known as a

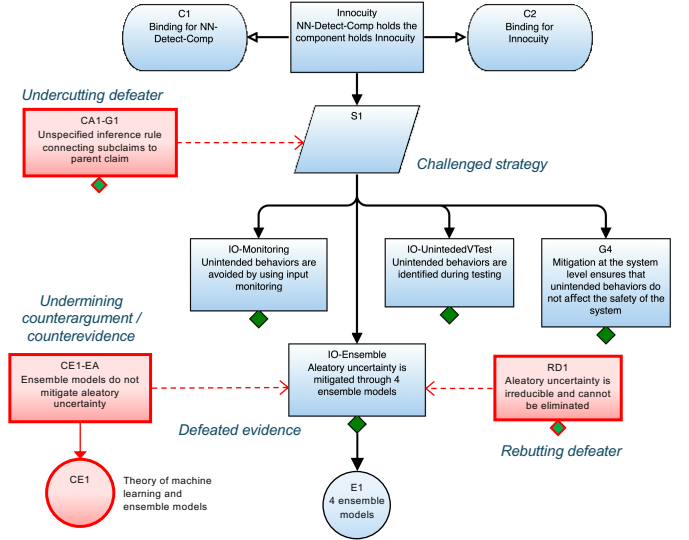


Fig. 2. Examples of argument defeaters and counterarguments in GSN.

counterargument. Following the terminology of [18], counterarguments can *rebut* claims, *undercut* strategies (i.e., inference rules), and *undermine* evidence (also see Section II-C). We consider the root claim of a counterargument to be an argument *defeater*, while *counterevidence* is evidence obtained in support of defeaters.

Fig. 2 gives an example of defeaters and counterarguments to an inductive argument from [19], whose main claim is that a neural network component, NN-Detect-Comp, “holds the component holds innocuity” [sic] (i.e., any non-required behavior does not lead to unacceptable safety impact) based, in part, on the evidence that four ensemble models mitigate aleatory uncertainty.² Apart from the obvious misstatement of the root claim, that argument is neither strong nor cogent, respectively, because (also see Section III-B3 and Section III-C2b):

- the inference rule(s) connecting the sub-claims and the evidence invoked to the property of the root claim, and the justification for that inference are both unstated (shown as the undercutting defeater in Fig. 2). Thus, there is poor support for the relevance of the sub-claims and the evidence invoked, and, in turn, for the main claim;
- one of the evidence items is neither *relevant* nor *credible* respectively: aleatory uncertainty is irreducible, challenging the assertion that it is mitigated (shown as the rebutting defeater in Fig. 2), and the assertion is also false based on the undermining counterevidence that ensemble models reduce epistemic rather than aleatory uncertainty.

Thus, identifying and eliminating defeaters and counterarguments assists in developing confidence in arguments, and their containing assurance case, in much the same way as confidence can be gained in system safety through the process

²The original argument in [19] is in natural language, structured according to a specific narrative template. Fig. 2 faithfully reproduces its content, including identifiers, conclusions, reasoning, premises, and any omissions or errors, although the original argument elements and the template have been mapped to the corresponding GSN syntax.

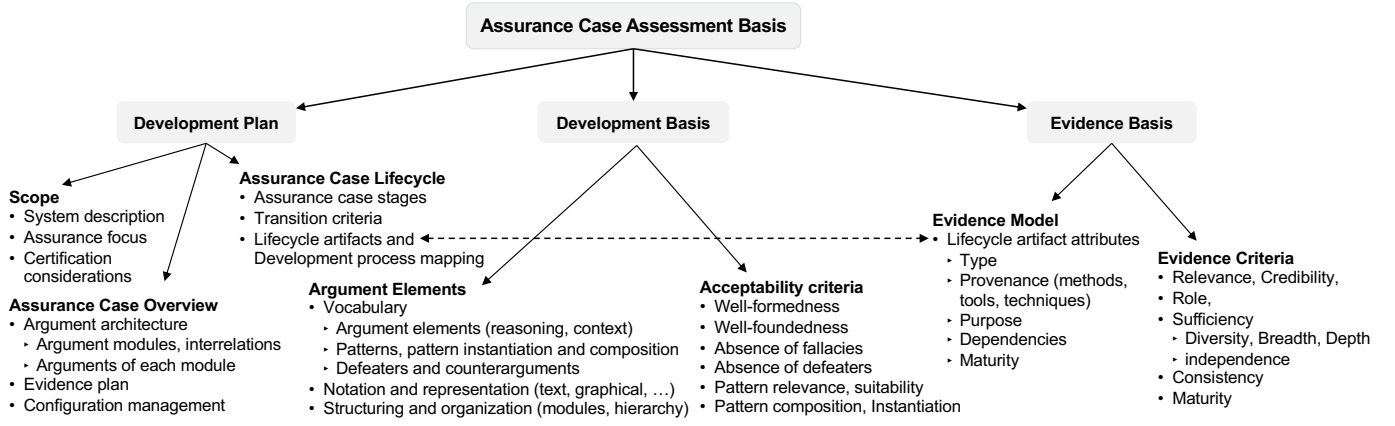


Fig. 3. Structured documents of the ACAB, their minimum proposed content, and their interrelations (dashed bidirectional arrow; not all shown).

of hazard identification and mitigation. As such, it is a key facet of the proposed assessment methodology (Section IV).

C. Evidence

Evidence refers to one or more lifecycle artifacts (i.e., the products of a process, method, or tool) accompanied by verifiable *evidence assertions*, i.e., qualitative or quantitative statements of fact, that directly or indirectly provide confirmation that a claim in an assurance argument can be considered to be true, e.g., that a product or service meets its requirements (and, therefore, can be relied upon). An evidence assertion is thus an attribute of a lifecycle artifact, rather than the artifact itself, and serves to link that artifact to the argument where it is invoked as evidence. For instance, the claim labelled IO-Ensemble in the argument of Fig. 2 is an evidence assertion, while the node labelled E1 (known as a *solution* in GSN) points to the specific lifecycle artifacts leading to that assertion. The combined structure represents the evidence for the main claim (the node labelled Innocuity in Fig. 2).

III. ASSURANCE CASE ASSESSMENT BASIS

Fig. 3 shows the three core structured documents comprising an assurance case assessment basis (ACAB), namely: a development plan (Section III-A), a development basis (Section III-B), and an evidence basis (Section III-C). These identify the arguments and their patterns that an assurance case will contain, clarifying the vocabulary and rules to describe those arguments, and declaring what evidence is necessary to substantiate the claims made. Additionally, suitability criteria for the arguments and evidence are specified.

Together, they provide a set of building blocks that: (i) a certification applicant may use to create their approach to develop an assurance case, and (ii) can be used to iteratively and systematically assess the resulting assurance case, towards facilitating a consensus between an applicant and the regulator.

A. Assurance Case Development Plan

1) *Purpose*: A development plan is intended to serve two main purposes. First, for a certification applicant, it establishes a roadmap and lifecycle for developing a credible assurance

case for an aeronautical system or service, such that there is a level of confidence commensurate with its safety criticality. Secondly, for both the certification applicant and the regulator, it serves as a valid, mutually accepted reference for the scope, content, lifecycle, and acceptance criteria of the assurance case to be produced. A key goal here is to avoid surprises during certification/approval review processes by providing confidence that an appropriate assurance case development process was consistently applied.

2) *Main Content*: A development plan (Fig. 3) comprises at least the following information: assurance case scope, lifecycle, and overview.

a) *Assurance Case Scope*: This is characterized by a system description, an assurance focus, and the relevant certification considerations. The system description, in turn, gives the necessary contextual information for the assurance case, such as: the system concept, the intended operating environment, a concept of operations in that environment including the intended usage and usage scenarios, and a conceptual architecture. The latter includes system functions, a high-level functional architecture including the relevant interfaces between the functions, as well as between the functions and their environment, and where appropriate allocations of the functions to physical equipment. The functional architecture may also include all or a part of the safety architecture [20] (Section II-A). For modifications or improvements to existing systems, the scope also includes descriptions of the changes and their potential impact on the rest of the system.

The assurance focus gives the main system characteristics in which confidence is necessary, e.g., the intended functionality and related dependability attributes such as safety or security. For ML-based functionality, for instance, the assurance focus can include a variety of related attributes including reliability, generalization, robustness, and stability.

Certification considerations identify any applicable regulations, regulatory guidance, and means of compliance.

b) *Assurance Case Lifecycle*: This describes a (sequence of) stages corresponding to concrete increments of the overall assurance case that will be developed. The lifecycle description

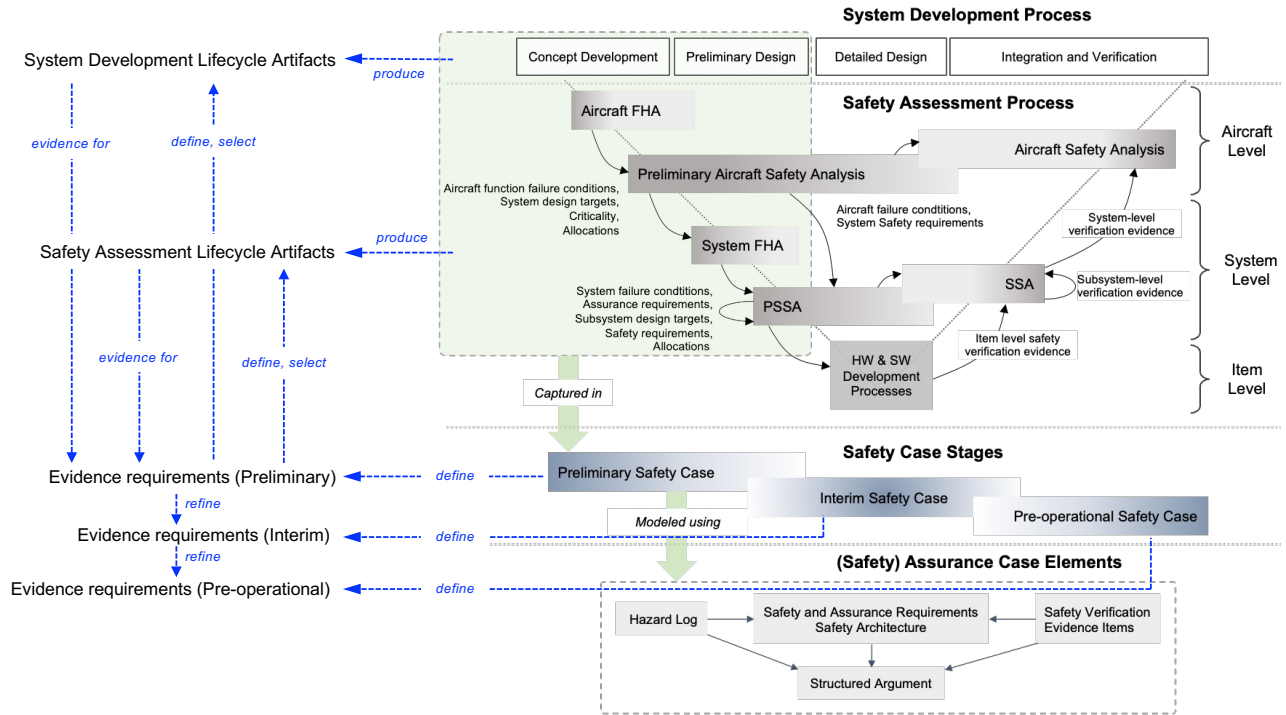


Fig. 4. Notional safety assurance case lifecycle stages related to system development and safety assessment processes, and the resulting lifecycle artifacts.

also includes completion or transition criteria, which serve as a quality gate to control the exit from and entry to the stated assurance case stages. The rationale for a staged assurance case lifecycle is to: (i) facilitate an incremental development and assessment of the assurance case; and (ii) ensure that the assurance case and the target system are concurrently developed to be mutually consistent, with the respective processes informing each other.

Fig. 4 depicts a notional, three-stage assurance case lifecycle showing different high-level relations between each stage and both the system development and safety assessment processes. The lifecycle artifacts that each process produces are invoked by the arguments in the corresponding assurance case stages, intuitively reflecting the provision of graduated confidence through a depth and breadth of evidence for the claims made in those stages.

Note that Fig. 4 is meant to be illustrative, and Section II-B1 clarifies that an assurance case may not be required or necessary when using standardized processes that have been accepted as a means of compliance for certification of a *conventional* system (i.e., one *not using* new and novel technologies such as ML). However, for example, when developing ML-based functionality for use in an aircraft, the idea is that a similar such mapping will link both the (non-standard) learning process chosen and its associated lifecycle data to the arguments and evidence of the assurance case to be used during approval or certification.

c) *Assurance Case Overview*: As a key component of the development plan, the assurance case overview summarizes the arguments to be developed and the evidence to be

invoked. As such, this overview is closely related to the two previous development plan components. Specifically, the scope constrains the applicability and relevance of the arguments and evidence; the assurance case lifecycle helps determine which evidence is necessary and for which arguments, where that evidence is obtained, and the sequence of argument development and evidence acquisition (Fig. 4).

The assurance case overview includes, at a minimum:

- an argument architecture (Section II-B4) potentially in the form of a module view or as a composition of patterns. For the former, the individual arguments that each module contains are also clarified at a high level, e.g., in terms of their main claims, supporting evidence assertions, and associated lifecycle artifacts;
- an *evidence plan* for gathering, curating, and managing the lifecycle artifacts declared in the argument architecture. This includes any relevant quantitative evidence, e.g., in the form of metrics, measures, and indicators [21]. The evidence plan may collate other related plans, e.g., for measurement and verification;
- a clarification of the configuration management considerations, including how consistency will be maintained between the system and its assurance case through their respective development lifecycles.

Additionally, for functionality that uses a mixture of novel and conventional technologies, the assurance case overview clarifies how the assurance case to be developed will support, or will be supported by, development assurance. For instance, consider a *hybrid solution* for an airborne collision avoidance function, a part of which uses ML models [22]. An assurance

case overview for that function will clarify how the arguments and evidence to be developed for the machine learnt portion relates to the confidence provided by existing software assurance practices [11] applied to the conventional software portion of that function.

B. Assurance Case Development Basis

1) *Purpose*: An assurance case and its underlying arguments can be presented in various forms (Section II-B1). Since there are no (aviation) industry consensus-based standards or guidelines for the same, the development basis has been conceived of as analogous to an *internal standard*, such as for software development, as invoked in airborne software assurance guidelines [11].

As such, the development basis is meant as the guideline a certification applicant can formulate to systematically develop their assurance case (in particular its arguments) in a consistent manner. Accordingly, it gives more details about the internal structure and elements of the arguments and patterns to be used, than the high-level overview in the development plan, which, instead, considers the structure that relates the different arguments of the assurance case. From an assessment standpoint, the development basis communicates to the regulator, the assurance case development guideline that the applicant intends to follow.

2) *Main Content*: The development basis specifies what constitutes the minimum content of acceptable assurance arguments together with rules for their development based on (but not limited to) the following considerations: vocabulary, representation and notation, structuring and organization, and acceptability criteria. These also apply to patterns and counterarguments, though each has specific, additional considerations.

a) *Vocabulary*: Arguments generally including reasoning and contextual elements. Reasoning elements comprise a root claim (i.e., a conclusion) linked by inference steps to (a chain of) sub-claims, some of which may be evidence assertions (Section II-B1), i.e., premises. Their semantics are grounded in contextual information, e.g., supporting assumptions, justifications, and references to relevant domain knowledge such as the content referenced in the scope and lifecycle elements of the development plan (Section III-A).

Claims may be further characterized by: (i) their *type*, that is, how they are presented, e.g., as categorical, probabilistic, empirical, or causal statements; and (ii) their assurance focus (Section III-A2a), e.g., safety, correctness, completeness, consistency, or coverage. For instance, a safety claim given as a probabilistic statement is: *the probability that the machine learnt component does not detect an object on the runway is no greater 3×10^{-6}* . Accordingly, various kinds of generic or application-specific argument patterns may be necessary, e.g., as in [13], [14].

b) *Representation and Notation*: Arguments (as well as patterns and counterarguments) may be presented in a number of ways including as text, tables, graphical notation, or their combinations (Section II-B1). The argument representations and notation chosen are to be applied consistently.

c) *Structuring and Organization*: Structuring mechanisms such as modularity (Section II-B3) may be required not only to manage argument size and improve comprehensibility, but also to organize them according to the assurance case overview (Section III-A2c). These considerations also apply to counterarguments, and compound arguments resulting from pattern composition and instantiation.

3) *Acceptability Criteria*: A broad but non-exhaustive set of criteria to be considered for accepting arguments, counterarguments, and patterns include:

a) *Well-formedness*: The stated vocabulary and notation are properly used with the correct structure, e.g., claims are declaratively stated, each conforming to their required type and assurance focus, and are connected to sub-claims or evidence assertions by inference rules that may themselves be justified by backing arguments.

b) *Well-foundedness*: The reasoning elements of an argument are examined for the following considerations:

(i) *Suitability of Evidence Assertions*: Evidence assertions, which may themselves be abstractions of sub-arguments, are both: *relevant* (directly or indirectly related to their higher level claims and the root claim) and *credible*, i.e., likely to be true, grounded in the evidence basis (Section III-C), and supported by the correct domain-specific facts, e.g., as invoked through valid contextual information declared in the scope (Section III-A2a).

(ii) *Relevance and Sufficiency of Inference*: Given the claims, sub-claims, and evidence assertions involved, and their respective type and assurance focus, the connecting inference rules invoked are suitable, and do not introduce circular dependencies³. Additionally, based on those inference rules, the evidence assertions or sub-claims individually or collectively provide support for their higher-level claims, and the root claim. This is assessed as *validity* for deductive arguments, and as *strength* for inductive arguments.

c) *Absence of Fallacies*: Arguments do not invoke known patterns of invalid or weak reasoning. For example, *hasty generalization* is an informal fallacy of weak induction where results drawn from a small or unrepresentative data sample is used as evidence for a broader, more general claim about a (statistical) population.

d) *Absence of Defeaters*: Defeaters and counterarguments to the argument elements (Section II-B5) have been identified (e.g., based on the preceding criteria), and addressed so that they have been either eliminated or accepted as *residual doubts* (see Fig. 5).

The following additional criteria are considered for patterns:

e) *Pattern Relevance and Suitability*: The types and assurance focus of the claims in a pattern conform to those of the claims, and associated evidence assertions declared in the assurance case overview. For example, a *claim formalization* pattern [14] is appropriate for a correctness claim stated as a logical proposition evidenced by formal verification results.

³Patterns may include circular dependencies as *loops*, with additional rules on both inference structure in the pattern, and the data used for instantiation, so that both the pattern and its instance arguments are well-founded [14].

f) *Admissibility of Composition*: Patterns can be properly composed, e.g., the leaf claim of a source pattern, the root claim of a target pattern, and their respective contextual elements, are compatible and consistent in a sequential composition [15].

g) *Pattern Instantiation*: The rules, data required, and constraints for pattern instantiation are specified.

C. Assurance Case Evidence Basis

1) *Purpose*: Similar to the development basis, the evidence basis is also meant as an internal guideline that an applicant formulates and communicates to the regulator. However, the intent is to establish and serve as an agreed upon reference for the evidence necessary in (the arguments of) the assurance case, such that its claims can be accepted with the necessary level of confidence.

The evidence basis is thus closely linked to the overall assurance case development plan (Section III-A): first, to the evidence generating processes to which the assurance case lifecycle is related (Section III-A2b); and, secondly, to the evidence plan formulated in the assurance case overview (Section III-A2c). Once created, the evidence basis houses the lifecycle artifacts invoked in the arguments of the assurance case, for which the development basis provides the necessary vocabulary, representation, and notation (Section III-B2).

2) *Main Content*: As shown in Fig. 3, the evidence basis comprises (at least) an evidence model, and evidence criteria.

a) *Evidence Model*: This clarifies what constitutes evidence for the planned assurance case, describing key attributes, interrelations, and candidate methods for evidence generation and verification. Such a model can then also be used to characterize previously available evidence (e.g., from the usage history of a earlier system variant, or a system of an equivalent type), and evidence that may be planned to be obtained (e.g., the performance of ML-based functionality operated in a constrained, and minimal risk environment, to support a broader deployment decision).

Candidate attributes of evidence include but are not limited to their: (i) *type*, e.g., experimental data, or the results of analysis; (ii) *provenance*, giving the source of the evidence, including processes, activities, methods, or tools used for its production; (iii) *version*, including details of configuration control; (iv) *purpose*, specifying the role in confirming, reinforcing, or otherwise opposing or weakening other items of evidence; (v) *assertions*, stating exactly what will be invoked in an argument, giving the qualitative or quantitative facts that can be directly established from a lifecycle artifact; (vi) *dependencies*, i.e., relations such as being created from, being a part of, or requiring other items of evidence; and (vii) state of *maturity* in a lifecycle, e.g., whether the evidence is required, obtained, verified, or accepted.

b) *Evidence Criteria*: Based on the attributes of the lifecycle artifacts in the evidence model, the evidence criteria constitute the rules to determine which of those artifacts are suitable, for which arguments, and thus which are admissible

as evidence for supporting the claims in an assurance case. Considerations for evidence criteria include at least:

(i) *Relevance and Credibility*: This assesses if a lifecycle artifact is acceptable as evidence in an argument, first, by examining whether it is semantically related, and then by examining the grounds for the same. Thus, evidence may be *direct* (semantically closer and thus relevant), *indirect* (semantically farther but still relevant), or unrelated (therefore *irrelevant*). Relevant evidence is credible when it can be corroborated, e.g., by appeal to the axioms or known truths of a well-defined domain, or through additional reasoning, e.g., as *evidence arguments* [23]. Note that this criterion is related to the well-foundedness criterion for argument acceptability (Section III-B3b). However, that considers whether the evidence assertions invoked in an argument are suitable for its higher level claims. Instead, here the purpose is evidence selection, to link a lifecycle artifact and its assertion attribute to the correct arguments and their respective evidence assertions (analogous to pattern composition).

(ii) *Role*: This determines whether a lifecycle artifact serves to confirm, reinforce, or weaken other evidence items or one or more claims in an argument. Whereas confirming evidence may be relevant for an argument, weakening evidence is instead relevant for the defeaters and counterarguments to an argument. The evidence model records the role as an attribute.

(iii) *Sufficiency*: This examines if the evidence provides the required level of support (or confidence) when used in an argument. Related aspects of this criterion include *diversity* (whether the same assertion can be made by lifecycle artifacts with different provenance), *breadth and depth* (quantity of evidence over and above diversity), and *independence* (whether evidence that separately confirms or reinforces other evidence is connected in terms of the source and method of production). Evidence that can be independently reproduced is preferable. When evidence cannot be independently reproduced, then that which can be independently checked is preferred.

(iv) *Consistency*: This relates to the previous criterion, in particular to the diversity and independence aspects: consistency of diverse and independent evidence assertions checks for a lack of mutual contradictions, providing stronger support for a claim whilst also increasing its credibility.

(v) *Maturity*: This indicates the progression of evidence in its readiness for use. As with the role, the evidence model records maturity as an attribute.

IV. ASSURANCE CASE ASSESSMENT FOR CERTIFICATION

This section gives a brief, and high-level clarification of how the ACAB and its document types can support assurance case assessment, inspired by current certification/approval review processes: broadly, applicant plans are reviewed and accepted; then their processes, which are based on accepted means of compliance, and the associated lifecycle artifacts are sampled to confirm conformity to the accepted plans.

Accordingly, there are two steps to the assessment method (Fig. 5): The first step is an *ACAB review* whose goal is to establish whether the ACAB itself is *acceptable*. That is, given

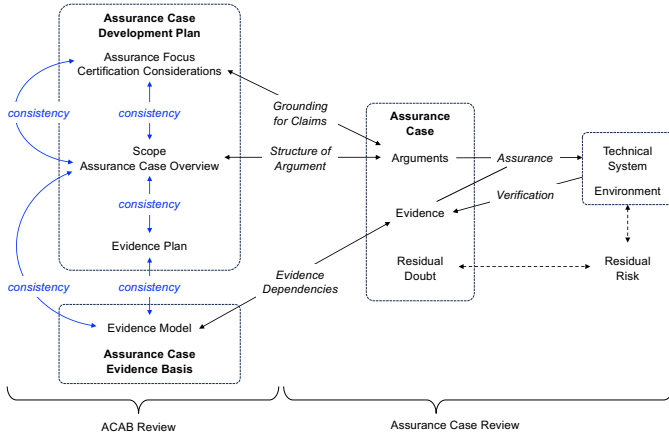


Fig. 5. Assurance case assessment methodology and consistency checking

the content of the ACAB documents, an acceptable ACAB provides confidence that the relevant conclusions of the to-be-developed assurance case may indeed be drawn with sufficient confidence. The second step is an *assurance case review*, whose goal is to check that the assurance case itself conforms to the acceptable ACAB.

The main focus of the ACAB review is the internal and mutual consistency of its constituent documents, which Fig. 5 shows as the corresponding consistency relations. Subsequently, other general quality attributes (e.g., comprehensibility of the assurance case overview and the assurance case development plan) are additionally evaluated.

Fig. 5 also shows some (but not all) of the consistency relations that will be assessed during the second step of assurance case review. Although this step is primarily focused on confirming that the assurance case conforms to the ACAB, it also involves evaluating the arguments and evidence according to their corresponding criteria (Section III-B3 and Section III-C2b). As indicated earlier, the identification of defeaters and counterarguments, followed by their subsequent mitigation or removal is a key criterion. However, not all defeaters may be removed, and thus are treated as *residual doubts* [4] that may be accepted with suitable justification. Note that assurance case review can involve multiple, pre-defined iterations (not shown in Fig. 5) which may be aligned with the stages of the assurance case lifecycle, or according to a broader certification plan.

A successful first step, i.e., an acceptable ACAB, can be seen as providing “credit” for the second step. Thus, the more detailed the assessment basis and its constituent planning elements—clearly and comprehensively articulating the assurance goals, how they will be achieved, and the substantiating evidence for the same—the more likely it is that consensus may be achieved on the acceptability of the assurance case, with reduced review effort (in particular, for the regulator performing the assessment).

An intent of the ACAB review is to manage project risk so that the subsequent assurance case review efforts may be directed towards auditing (the applicant process and lifecycle

artifacts), rather than detailed technical assessments of the arguments and evidence. This is not to say that the latter will be unnecessary during assurance case review, rather that fewer technical assessments may be needed by the regulator, and that such assessment effort may instead be focused on those aspects of the arguments and evidence that the regulator has deemed important for independent assessment.

V. EXAMPLE: HIGH ALTITUDE PLATFORM SYSTEM

We use an example of a high altitude long endurance (HALE) unmanned aircraft system (UAS), also known as a high altitude platform system (HAPS), to illustrate both the content of its assurance case assessment basis (ACAB), and the assessment methodology. Due to space constraints, note that the example is not comprehensive, and the main focus is on selected elements of: (i) the development plan and the evidence basis—which also intentionally contains some defects to highlight the assessment approach—and (ii) the ACAB review step of the assessment methodology, in particular evaluation of the assurance case overview element of the development plan. More details on this example are in [24], [25].

A. HAPS Assurance Case Development Plan Excerpts

1) *System Description*: The HAPS is a 35m wide, fixed wing, twin propeller, carbon fiber composite airframe weighing 150kg, powered by a combination of lightweight solar cells and re-chargeable electric batteries. It has an on-board autopilot system, and can also be controlled from a ground control station through redundant communication links.

2) *Operating Concept and Environment*: The aircraft operates in the stratosphere at a nominal altitude of 60000ft. The aircraft will be launched from and recovered in special use airspace (SUA) from where it accesses a specified transit corridor, to and from its mission area. The route comprises predefined waypoints, including safety waypoints for potential diversions.

3) *Assurance Focus*: The main assurance focus is *operational safety*: the aircraft must not pose a collision hazard to other airspace users, or to the population on the ground. A related assurance focus is *airworthiness*: the aircraft must be able to safely attain and maintain flight for an extended duration, follow its pre-determined route for transit and return transit, and be able change its flight path when commanded.

4) *Relevant Certification and Approval Considerations*: The requirements of the following regulatory guidance documents are potentially applicable: (i) US DOT, FAA Order 8040.6A, Unmanned Aircraft System (UAS) Safety Risk Management (SRM) policy, Sep. 2023, and (ii) US DOT, FAA Order 8110.4C CHG 7, Type Certification, Oct. 2023.

5) *Assurance Case Overview*: The overall assurance case (Fig. 6) is organized in two stages using a modular approach.

a) *Concept Safety Case*: This addresses the safety considerations of the operating concept, mission design, and platform airworthiness. The main claim is that the operating concept does not pose a risk greater than the acceptable target level of safety (TLOS) specified. That claim is supported by

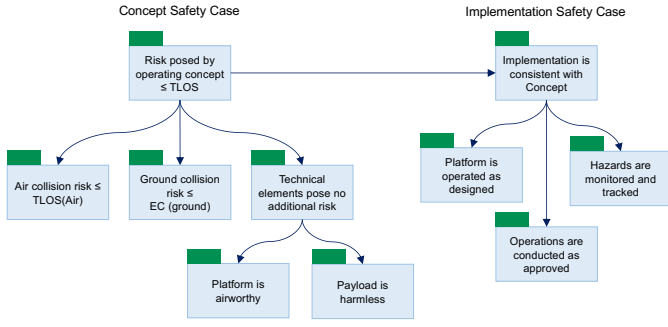


Fig. 6. Proposed argument architecture for the HAPS Safety Case

three complementary lines of argument whose main conclusions are that:

(i) *Air collision risk does not exceed the acceptable TLOS*: The supporting argument appeals to an analysis of the relevant historic air traffic data in the applicable airspace and for the applicable operating duration, for the entire flight profile to show that the probability of encountering conflicting air traffic is lower than the required TLOS.

(ii) *Ground collision risk does not exceed the Expectation of Casualty (EC) on the ground*: The supporting argument is by analysis of EC given the expected aircraft descent profile in the event of a loss of flight capability and the population densities of the areas overflown.

(iii) *The technical elements of the system pose no additional risk*: The supporting arguments are by demonstration of platform airworthiness, and that the payload is harmless.

b) *Implementation Safety Case*: This includes safety considerations pertaining to operational procedures, including general flight operations, and mission-specific operations. The associated main claim is that that the system will be implemented as stated in the concept safety case, for which, again, there are three complementary lines of supporting assurance arguments, i.e., that: (i) the platform is operated as designed; (ii) the operations are conducted as approved; and (iii) safety hazards are monitored and tracked.

B. HAPS Assurance Case Evidence Basis Excerpt

The following lifecycle artifacts will provide evidence for the air collision risk mitigation argument module:

- Launch and recovery area map and coordinates overlaid on a sectional aeronautical chart showing containment entirely within an SUA, with concurrence from the SUA authority for operational use of the aircraft;
- Primary and secondary surveillance radar tracks as per the following specification:
 - Air route traffic control centers: Z01, Z02, Z03;
 - Primary Route Waypoints: R01 ... R20;
 - Alternate Route Waypoints: A01 ... A10;
 - Safety Waypoints: S01 ... S10;
 - 3-dimensional airspace volumes: V01 ... V20;
 - for a 5 year period from 01/2020 for the months of January to May inclusive;

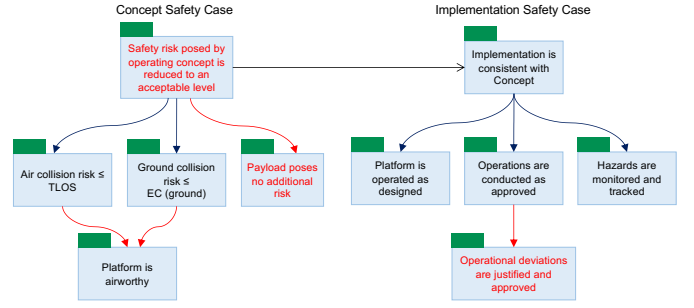


Fig. 7. Revised HAPS Safety Case Argument Architecture

- Statistical analysis of the above historical radar track data above to determine the air traffic patterns and density in the relevant airspace for the entire flight profile;
- Quantitative analysis of the air traffic radar tracks to determine collision risk in terms of collision rate;
- Seasonal weather data and analysis of weather patterns for the same time period and locations as covered by the radar track data.

C. Excerpt of Assurance Case Assessment Basis Review

This section gives an example result from the first step of the proposed assessment methodology, i.e., ACAB review. We focus on the review of the development plan, in particular its assurance case overview (Fig. 6). The corresponding argument architecture contains the following deficits:

(i) *Validity of the main claim of the concept safety case*: The TLOS concept is not appropriate for the aggregate (air and ground) safety risk posed by the system and its operations, and needs to be restated.

(ii) *Correctness of the relational structure of the argument modules in the concept safety case*: The arguments for air and ground collision risk mitigation are both dependent on platform airworthiness, which is not reflected in the argument architecture.

(iii) *Completeness of implementation safety case*: The argument for justified and approved operational deviations from the concept safety case if necessary is missing.

Fig. 7 shows the result of revising the argument architecture to correct the preceding deficits. Specifically, (i) the root claim of the concept safety case has been modified to reflect the more general claim of reducing of safety risk to an acceptable level; (ii) the dependency of the air and ground collision risk mitigation argument modules on the argument module of an airworthy platform has been indicated; and (iii) the argument module identified as missing in the implementation safety case, has now been included.

VI. CONCLUDING REMARKS

This paper has motivated the need for a systematic framework to support the development and review of assurance cases. Towards that end, it has outlined the primary elements of an approach centered on a notion of an assurance case

assessment basis (ACAB), comprising three structured documents, namely a development plan, a development basis, and an evidence basis. The intent is to provide an approach through which a certification applicant and regulator can reach consensus and explicitly document how a series of assurance cases will be developed, reviewed, and ultimately accepted.

An additional goal of the proposed approach is to be compatible with the prevailing certification and approval review processes. Based on that, an assessment methodology has been proposed that involves, first, reviewing the ACAB, followed by an assessment of assurance case conformity with the ACAB, including reviews of the underlying arguments and evidence.

The following parallels can be observed in comparison with current certification and approval review processes: the development plan can be viewed as analogous to setting forth the scoping, planning, and evidence elements of what currently constitute *acceptable means of compliance* (AMCs). However, whereas AMCs specify design and development standards produced from broad industry consensus, the assurance case development plan rather captures a certification applicant's high-level arguments why the evidence gathered from their defined/chosen process, according to their stated plans, and for the scope declared, may meet the regulatory requirements. Additionally, the development and evidence basis documents are analogous to the internal standards and guidelines invoked by current AMCs.

Though the proposed approach is intended to be notation-agnostic, it has been clarified using GSN arguments, with an illustrative example highlighting some selected ACAB content. Additionally we have presented excerpts of the results of assessing the assurance case overview of a high altitude platform system whose main assurance focus is operational safety, and airworthiness. We believe that this structured approach, which is intended to systematize best practices towards reducing the assessment burden, offers significant potential for automation. This could help both the applicant as they develop the assurance case, and the assessor as they check for conformance with the assessment basis.

As a next step, we plan to develop a detailed process and workflow to concretize the proposed assurance case assessment methodology. Additionally, towards validating its utility and applicability, we intend to apply it to assess arguments developed by others.

REFERENCES

- [1] US Department of Transportation, Federal Aviation Administration, "Type Certification," National Policy, Order 8110.4C Chg 7, October 2023. [Online]. Available: https://www.faa.gov/documentLibrary/media/Order/Order_8110.4C_CHG_7.pdf
- [2] D. Rinehart, J. Knight, and J. Rowanhill, "Current Practices in Constructing and Evaluating Assurance Cases With Applications to Aviation," NASA, Technical Report NASA/CR-2015-218678, January 2015. [Online]. Available: <https://ntrs.nasa.gov/citations/20150002819>
- [3] J. Rushby, X. Xu, M. Rangarajan, and T. Weaver, "Understanding and Evaluating Assurance Cases," NASA, Technical Report NASA/CR-2015-218802, September 2015. [Online]. Available: <https://ntrs.nasa.gov/citations/20160000772>
- [4] R. Bloomfield and J. Rushby, "Confidence in Assurance 2.0," *arXiv:2205.04522*, May 2022. doi: [10.48550/arXiv.2205.04522](https://doi.org/10.48550/arXiv.2205.04522).
- [5] E. Asaadi, E. Denney, J. Menzies, G. Pai, and D. Petroff, "Dynamic Assurance Cases: A Pathway to Trusted Autonomy," *IEEE Computer*, vol. 53, no. 12, pp. 35–46, 2020. doi: [10.1109/MC.2020.3022030](https://doi.org/10.1109/MC.2020.3022030).
- [6] E. Denney and G. Pai, "Assurance-driven Design of Machine Learning-based Functionality in an Aviation Systems Context," in *2023 IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC)*, October 2023. doi: [10.1109/DASC58513.2023.10311282](https://doi.org/10.1109/DASC58513.2023.10311282).
- [7] The Assurance Case Working Group (ACWG), "Goal Structuring Notation Community Standard Version 3," SCSC-141C, May 2021. [Online]. Available: <https://scsc.uk/r141C:1>
- [8] E. Denney and G. Pai, "AdvoCATE: The Assurance Case Automation Toolset at Age 14," in *AIAA SCITECH 2026 Forum*, no. AIAA 2026-1251. AIAA, January 2026. doi: [10.2514/6.2026-1251](https://doi.org/10.2514/6.2026-1251).
- [9] SAE S-18 and EUROCAE WG-63, "Guidelines for Development of Civil Aircraft and Systems," ARP4754B/ED-79B, Oct. 2023.
- [10] —, "Guidelines for Conducting the Safety Assessment Process on Civil Aircraft, Systems, and Equipment," ARP4761A/ED-135, Oct. 2023.
- [11] RTCA SC-205 and EUROCAE WG-71, "Software Considerations in Airborne Systems and Equipment Certification," DO-178C/ED-12C, Dec. 2011.
- [12] SAE G-34 and EUROCAE WG-114, "Artificial Intelligence in Aeronautical Systems: Statement of Concerns," AIR6988/ER-022, April 2021.
- [13] R. Bloomfield and K. Netkachova, "Building Blocks for Assurance Cases," in *2014 IEEE International Symposium on Software Reliability Engineering Workshops*, 2014, pp. 186–191. doi: [10.1109/ISSREW.2014.72](https://doi.org/10.1109/ISSREW.2014.72).
- [14] E. Denney and G. Pai, "Safety Case Patterns: Theory and Applications," NASA Ames Research Center, Technical Report NASA/TM-2015-218492, Feb. 2015. [Online]. Available: <https://ntrs.nasa.gov/citations/20150004086>
- [15] —, "Composition of Safety Argument Patterns," in *Computer Safety, Reliability and Security. SAFECOMP 2016*, LNCS vol. 9922, A. Skavhaug, J. Guiochet, and F. Bitsch, Eds. Springer, Sep. 2016. doi: [10.1007/978-3-319-45477-1_5](https://doi.org/10.1007/978-3-319-45477-1_5).
- [16] —, "Towards a Formal Basis for Modular Safety Cases," in *Computer Safety, Reliability and Security. SAFECOMP 2015*, LNCS vol. 9337, F. Koornneef and C. van Gulijk, Eds., Springer, Sep. 2015. doi: [10.1007/978-3-319-24255-2_24](https://doi.org/10.1007/978-3-319-24255-2_24).
- [17] —, "Architecting a Safety Case for UAS Flight Operations," in *34th International System Safety Conference (ISSC)*, Aug. 2016.
- [18] J. Goodenough, C. Weinstock, and A. Klein, "Eliminative Argumentation: A Basis for Arguing Confidence in System Properties," Carnegie Mellon University, Software Engineering Institute, Technical Report CMU/SEI-2015-TR-005, Feb 2015. doi: [10.1184/R1/6573413.v1](https://doi.org/10.1184/R1/6573413.v1).
- [19] H. Späth, T. Varchev, S. Staudacher, Z. Daw, and M. Holloway, "Arguing Machine Learning Assurance for Certification," in *Software Engineering 2025-Companion Proceedings*. Gesellschaft für Informatik, 2025, pp. 101–120. doi: [10.18420/se2025-ws-09](https://doi.org/10.18420/se2025-ws-09).
- [20] E. Denney, G. Pai, and I. Whiteside, "The Role of Safety Architectures in Aviation Safety Cases," *Reliability Engineering & System Safety*, vol. 191, 2019. doi: [10.1016/j.res.2019.106502](https://doi.org/10.1016/j.res.2019.106502).
- [21] E. Denney and G. Pai, "Reconciling Safety Measurement and Dynamic Assurance," in *Computer Safety, Reliability, and Security. SAFECOMP 2024*, LNCS vol. 14988, A. Ceccarelli, M. Trapp, A. Bondavalli, and F. Bitsch, Eds. Springer, 2024, pp. 51–67. doi: [10.1007/978-3-031-68606-1_4](https://doi.org/10.1007/978-3-031-68606-1_4).
- [22] M. Damour, F. De Grancey, C. Gabreau, A. Gauffriau, J.-B. Ginestet, A. Hervieu, T. Huraux, C. Pagetti, L. Ponsolle, and A. Clavière, "Towards Certification of a Reduced Footprint ACAS-Xu System: A Hybrid ML-Based Solution," in *Computer Safety, Reliability, and Security. SAFECOMP 2021*, LNCS vol. 12852, I. Habli, M. Suján, and F. Bitsch, Eds. Springer, 2021, pp. 34–48. doi: [10.1007/978-3-030-83903-1_3](https://doi.org/10.1007/978-3-030-83903-1_3).
- [23] E. Denney and G. Pai, "Evidence Arguments for Using Formal Methods in Software Certification," in *2013 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, Nov. 2013, pp. 375–380. doi: [10.1109/ISSREW.2013.6688924](https://doi.org/10.1109/ISSREW.2013.6688924).
- [24] E. Denney and G. Pai, "Assurance Case Assessment Basis", US Dept. of Transportation, Federal Aviation Administration, Technical Report DOT/FAA/TC-, May 2025. (to appear)
- [25] E. Denney and G. Pai, "Assurance Case Assessment and Review – Methodology and Workflow", US Dept. of Transportation, Federal Aviation Administration, Technical Report DOT/FAA/TC-, April 2026. (to appear)