

Computer-Aided Evaluation for Argument-Based Certification

Zamira Daw*, Timothy Wang*, Chanwook Oh[†], Matthew Low[†], Isaac Amundson[‡],
Guoqiang Wang*, Ryan Melville*, Pierluigi Nuzzo[†]

*Raytheon Technologies Research Center, Berkeley, CA, USA

[†]University of Southern California, Los Angeles, CA, USA

[‡]Collins Aerospace, Minneapolis, MN, USA

Abstract—Next-generation certification processes are expected to be influenced by two dominant trends: customizable certification, as advocated by Overarching Properties (OPs), and the continuous and rapid delivery approach due to the coupling of software development and operations (DevOps). In light of these trends, certification processes must adapt to support the ongoing evaluation of assurance while accommodating the emergence of new evidence and evolving safety and security practices, particularly for systems enabled by artificial intelligence (AI). These requirements call for the development of computer-aided evaluation tools that automate repetitive tasks and offer visualization aids to assist evaluators and applicants in making informed decisions. This paper aims to address the challenges associated with assurance evaluation via a survey-based analysis and proposes a user interface that streamlines the evaluation of certification arguments.

Index Terms—Certification, Argument-based certification, Human-Machine Interface

I. INTRODUCTION

Certification processes in the aerospace industry are paramount for ensuring the quality and safety of aerospace products. However, the rapid development and widespread adoption of new technologies have presented challenges in keeping certification standards up to date. Recognizing this issue, government agencies have acknowledged the need to adapt certification processes to effectively address the ever-evolving technological landscape.

Currently, applicants have the opportunity to propose innovative means of compliance with airworthiness standards, but there is no guarantee of their approval. To overcome this challenge, NASA and the FAA have actively supported the development of the Overarching Properties (OPs) concept. OPs introduce three high-level fundamental properties that serve as the foundation for proposing novel means of compliance. This approach allows greater flexibility for applicants in certifying

their systems, enabling them to present customized arguments to demonstrate compliance. Such flexibility is particularly crucial for artificial intelligence (AI) systems, where adaptability to changing technologies is paramount. Embracing this approach can make certification processes more agile and better equipped to handle the certification challenges posed by emerging technologies.

Another important trend in the industry is the coupling of software development and operations, known as DevOps, aimed at continuous delivery with high software quality. This coupling has extended to security aspects, referred to as DevSecOps, to integrate security practices continuously. Although the certification process can benefit from DevSecOps by extracting automatic assurance evidence of the system, the evaluation process remains predominantly static and reliant on human-driven processes. To help address this issue, DARPA has initiated the Automated Rapid Certification Of Software (ARCOS) program, which aims to automate the evaluation of software assurance evidence, enabling rapid determination of system risk acceptability by certifiers. A certification process that continuously evaluates system assurance as new evidence is created and new safety and security practices are discovered is highly desirable.

Argument-based certification is an innovative approach that effectively tackles both of these trends. It utilizes certification arguments as a structured method to propose novel means of compliance, ensuring a clear understanding by certification authorities. Furthermore, this approach embraces digitalization and allows automatic accomplishment evaluation throughout the development and operation phases. By incorporating argument-based certification, the aerospace industry can benefit from a streamlined and comprehensive certification process that facilitates understanding and ensures compliance at every stage.

While significant efforts are underway in academia and industry to advance the development of argument-based certification [1]–[5] there remains a critical gap in addressing the human factors intrinsic to the certification process. The introduction of argument-based certification brings about a fundamental shift in the role of the evaluator, expanding their responsibilities from solely assessing compliance with predetermined certification standards to also evaluating the

Distribution statement “A” (approved for public release, distribution unlimited). This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA), contract FA875020C0508. The views, opinions, or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. The authors wish to also acknowledge the partial support by the National Science Foundation (NSF) under Awards 1846524 and 2139982, the Office of Naval Research (ONR) under Award N00014-20-1-2258, the Defense Advanced Research Projects Agency (DARPA) under Award HR00112010003, and the Okawa Research Grant.

acceptability of customized and innovative means of compliance. Consequently, there arises a pressing need for tools that can provide comprehensive support to human evaluators. Such tools should automate specific tasks and offer intuitive visualization aids to enable evaluators to make informed approval decisions confidently. By addressing these human factors, the certification process can become more efficient, effective, and adaptable to the evolving landscape of aerospace technologies.

This paper serves as an exploration of the challenges surrounding argument-based certification, with a specific focus on the human factors involved. To shed light on these issues and propose potential solutions, we conducted a survey that involved four certification experts from three distinct companies. Building upon the findings of our survey, we have developed a user interface called EVAL tool, which is designed to facilitate the evaluation of certification arguments. EVAL tool is designed with the purpose of streamlining the argument-based certification process and empowering evaluators to make well-informed decisions within the context of a complex certification environment.

II. STATE OF THE ART

An *assurance case* (AC) is an argument constructed to establish that a system meets the requirements in its operating environment. It usually follows a hierarchical approach, where a *claim* is supported by *evidence* through strategies and intermediary claims [6], [7]. ACs are often described using structured languages or graphical notations like the Claims-Arguments-Evidence (CAE) notation [8] and the Goal Structuring Notation (GSN) [9]. An AC can be visualized as a directed acyclic graph, with the system specification (the top-level claim) represented at the root node and the evidence at the leaf nodes. Several studies have demonstrated the potential benefits of using arguments for the certification process of future systems [10], [11].

Several tools [12]–[15] are available to assist with the creation, instantiation, management, and analysis of ACs, either through manual or partly automated processes. Approaches based on the AMASS platform [16]–[19] utilize a contract-based approach to automate the creation of ACs, enhancing the efficiency and effectiveness of the AC development process via the compositionality of contracts and reuse of argumentation patterns. While there are a few tools available, such as ExplicitCase [20] and the methodology proposed by Ramakrishna et al. [21], that support automated AC generation and confidence assessment, these tools do not operate within a contract-based or compositional framework. Moreover, some elements of the semantics in these tools are not well-defined in their respective conventions, leaving room for an individual developer to clarify them [6]. The characterization of refinement steps between claims, whether they are inductive or deductive, often lacks rigor, creating a potential confirmation bias that may influence the analysis [7]. Our approach focuses on leveraging contract operations to establish solid relationships

between claims and integrating Bayesian reasoning techniques to evaluate the strength of such claims.

Finally, most of the existing efforts assume that certification authorities and applicants are already familiarized with the notations, such as GSN, employed to represent ACs. On the other hand, EVAL tool facilitates the evaluation of certification arguments by providing multiple options for representing an AC. Users, such as current practitioners, can choose to represent an AC using a combination of graphical, textual, and tabular representations. We use a color-coded graph to illustrate the overall structure of an AC and confidence in various claims within the AC. At the same time, textual and tabular representations are utilized to show the details in the argumentation steps and (sub-)claims.

III. ARGUMENT-BASED CERTIFICATION

Argument-based certification represents a departure from the traditional two-step human judgment process observed in certification. In the conventional approach, a group of experts establishes a set of objectives based on best practices, such as those outlined in standards like DO-178C, to ensure system assurance. Subsequently, a certification designee evaluates the evidence provided by the applicant to determine compliance with these objectives. However, the underlying rationale behind these best practices is often unclear within the standards themselves, and trust in them relies heavily on the consensus of experts. Consequently, adapting these practices to special cases can be challenging. Applicants carry out the required activities (e.g., requirement reviews, code standards, testing procedures), and provide the results of these activities to the certification designee as evidence of compliance.

Due to the voluminous nature of the evidence, the certification designee adopts a sampling approach, selecting a subset of high-level requirements and examining all the related evidence (e.g., reviews and traced test cases). This approach allows detecting systematic errors in the development process (e.g., insufficient questions in checklists) and gaining a better overview of the development process. However, the sampling approach does not ensure the absence of errors in the system. This is usually not a problem when companies diligently follow the proposed plans, which are thoroughly reviewed by the authorities.

In contrast, argument-based certification offers the applicants the opportunity to present a case for the sufficiency of their proposed means of compliance. The Overarching Properties (OPs) provide a framework for constructing these arguments based on three fundamental properties [22], offering guidance while allowing considerable flexibility for applicants. Consequently, certification authorities must not only assess the evidence provided but also evaluate the adequacy of the proposed means of compliance. This necessitates that certification designees have access to the relevant technology associated with the novel means of compliance, which is not currently a requirement in the conventional certification process. To mitigate the risk associated with the approval of means of compliance, it is crucial to present a preliminary certification

argument during the system planning phases [23]. At this early stage, the argument may not be fully complete and is subject to change as the development progresses. Nonetheless, the preliminary certification argument can offer valuable insights to the certification designee for preliminary assessment. Throughout the development process, evidence supporting the argument is generated and collected. Upon completion of the development process, the preliminary argument transitions into a final assessment argument, which encompasses the utilized means of compliance along with the supporting evidence.

IV. SURVEY

While there are a few tools for visualization of assurance cases (see Section II), these tools are not commonly used by system and safety engineers or certification authorities. Furthermore, certification in aerospace is an already well-established process, in which every certification authority has their own evaluation method. The transition to an argumentation-based certification approach can take time. Also, due to the efficacy of current certification process, established evaluation processes will still apply for classical systems. Therefore, we questioned certification authority representatives about the current evaluation process of the Plan for Software Aspects of Certification (PSAC) and the Software Accomplishment Summary (SAS) and the possible need for evaluating argument-based certification. This section presents the survey process, summarizes the questionnaire results, and identifies possible methodologies and tool features that could support the evaluation process.

A. Subjects

Four certification experts from three different aerospace companies were interviewed. Three experts are certification designees and one is a system engineer that is actively involved in the certification process.

B. Process

The survey was performed as an iterative process divided in three phases: Exploration, Features, and Usability. The first phase was focused on capturing the evaluation process through open-end questions that are currently used and that could be used for argument-based certification. In the Exploration phase, we developed a questionnaire with six open-ended questions. Participants only sent written responses to the authors, which ensured that their respective perspectives were not influenced by others. After analyzing the answers, the authors had one-on-one sessions with the subjects to clarify their answers and avoid any misinterpretation. During the same session, we presented a high-level analysis of the collective answers, to which the subjects provided their perspective.

For the Feature phase, we developed a mock-up tool based on the survey results and asked close-ended questions about the possible methodologies and tool features that can facilitate informed approval decisions from an evaluator, and increase the trust in the use of automation within a certification process. For the Usability phase, we presented to the subjects the

implemented tool and asked close-ended questions about the usability of the tool.

C. Survey Questions

Context:

It is the first time that you receive assurance cases from an applicant to show means of compliance using Overarching Properties.

Question 1:

What needs to be done to dramatically simplify or streamline certification tasks that certification experts could face daily in the context of OPs (keep in mind that different types of arguments can be proposed by every applicant)?

Question 2:

What does the ideal certification tool look like for you?

Question 3:

How would a certification expert like to see assurance cases presented to you?

Question 4:

Considering that assurance cases may be very large, we explore using different abstractions.

- a What abstraction level would be the most useful for understanding and evaluating the argument?
- b When details are needed, how would you prefer to see them?

Question 5:

What brings you confidence in an assurance case?

Question 6:

Would you trust its accompanying confidence value? Or what information is required to increase your trust in an assurance case?

D. Results

1) *Assurance Patterns and Assurance Case:* In an argument-based certification environment, arguments need to be structured in a standardized manner for clarity and efficiency. This can be supported by offering a library of standardized argumentation patterns. Instantiated with a specific context, argumentation patterns can be used to automatically generate assurance cases that provide clarity as to the detailed plans, procedures, review forms, work instructions, verification reports, configuration reports, quality assurance reports, etc. An assurance case needs to provide all artifacts that are needed to substantiate the argument such that the designees or FAA engineers can conclude that the project should be granted for approval.

Using assurance cases leads to a new paradigm. It should enable viewers to click on a “link” in the case to access underlying supporting data and see what is behind it during the certification project development and audit or interview threads. The complete set of data used in assurance cases needs to be shown in a consistent manner. All objectives need

to be backed up with intermediate evidential arguments and ultimately with “leaf-level” concrete evidence.

2) *Visualization*: Visualization is a key element of advanced tools for evaluating argument-based certification. For certification development engineers, it is about how to better organize applicants’ data to support the sought approval for their products. For auditors, the tool is mostly a way to view the data that the applicant is providing.

The tool needs to address issues related to presentation format (graphical versus textual) and abstraction levels for maximal effectiveness. Graphics and text have their own ideal use case scenarios. Experience shows that a graphical representation works better in understanding an architectural view at a high level of abstraction while low-level details are better in the format of text. For example, graphics are preferred over text for displaying the high-level argumentation architecture and supporting artifacts are better represented in textual or tabular references.

Assurance cases are new in avionics software certification. There may exist many levels of justification from a top-level objective to bottom-level evidential artifacts. They need to be presented in a hierarchy based on levels of abstractions to address the complexity issue.

At the top level, the general structure of the justification is displayed. The tool needs to enable viewers (certification development engineers and compliance officers) to investigate a question by allowing them to drill down to the evidence that is used to support the claim of that item and continue to drill down in greater and greater detail to the leaf-level evidence, if needed. These navigation capabilities associated with the visualization of assurance cases will significantly improve efficiency for development engineers and compliance officers to spot check and verify evidence compliance.

Different views need to be supported by the argument-based certification environment. There are multi-fold benefits. A view on a particular aspect will remove unnecessary complexities from other aspects that are irrelevant to the focus and hence enables tractability. Another important use case is in providing technical and non-technical views. From time to time, potential customers and certification personnel may also be a competitor, e.g., a government or certification approval expert is employed by a competitor company. Such a perspective view offering can restrict only certain people during certain times and therefore helps protect intellectual property (IP) for the applicant.

3) *Documentation*: Documentation is an overloaded word. For a tool, it usually refers to materials describing how to use the tool. However, here we refer to documentation as the documents that are generated by the tool for its users.

For certification development engineers, the argument-based certification environment should automatically generate planning materials such as the PSAC. The tool needs to be able to automatically populate a report on the objectives met from the guidance documents like DO-178C.

For auditors, the tool needs to be able to automatically document any data compliance that is examined during an

audit or interview in a report and produce a comprehensive assessment report per job-aid.

4) *Version Control*: During the lifetime of a product, there may exist multiple versions of the product component implementations. The multiple versions may come from either bug fixes or support of new features. The argument-based certification environment needs to accommodate certifications for the product based on different versions of its component implementations. Given a component, the tool needs to be able to display all certification approvals of those products that use its different implementation versions.

5) *Feedback and Rating*: With the goal of simplification of the certification personnel’s tasks, the argument-based certification environment needs to continuously improve based on feedback from users. The users need to be able to rate the environment by sharing how their assessment of the artifacts matched the confidence value generated by the tool. The environment needs to offer an interface for users to upload and share their certification projects, which will ultimately enrich the set of available training and tutorial materials.

E. Acceptance and Confidence

Introduction of new tools such as the argument-based certification environment to a community may experience resistance. Even with a reasonable level of acceptance, confidence of users may still need to be enhanced. In the rest of this section, we discuss what can be done to promote acceptance.

1) *Training*: Experience tells that first-time users of a new tool will usually have little trust in the tool. This certainly applies to a new certification environment. When a new concept is introduced to the certification community, new accompanying formalisms are also advocated. For example, the overarching property-based certification regime is proposed as an alternative to current certification practices for more flexibility. It is crucial to make sure that users understand the difference between desired behavior and defined intended behavior and what it really means that the defined intended behavior is correct and complete with respect to the desired behavior. Users may need to change the reasoning process developed for their certification practices being performed for years.

To jump start and quickly ground certification personnel in the new environment, extensive in-depth tutorial materials need to be prepared based on execution of some pilot certification programs by the developers of the argument-based certification environment together with certification approval authorities. The tutorials need to focus on the rationale and confidence in the approaches that are offered by the new certification environment. It would be very helpful if the training presentation or tutorial is delivered by some credentialed experts in the community. Ideally, the training and tutorial need to demonstrate that the same tool is in use for multiple programs instead of a toy example. Notice that the training and tutorial materials can be natively incorporated as educational modules into the certification environment itself.

In addition, training workshops can be offered to the certification community. Either built-in tutorial modules or custom-prepared training materials or both can be used in such sessions. Lastly, support from the government certification agencies like the FAA and EASA will also help certification experts gain more confidence in the new certification environment.

Just like qualification of tools used for product development, validation and verification, the developers of the argument-based certification environment can have the tool itself be qualified by certification authorities, which will involve agencies like the FAA, EASA, and Transport Canada. Besides increasing user confidence, another benefit of this is that the certification package documents produced under a qualified tool can be less detailed, which further simplifies the tasks that face certification personnel.

V. EVAL TOOL

Based on the results of the survey, we developed a user interface for evaluating certification arguments, called EVAL tool. A certification argument is an assurance case (AC) that describes the means of compliance.

In the case summary dashboard of the EVAL tool shown in Figure 1, a table of the top ten AC candidates are displayed. The ranking of candidates is based on the following sorting scheme and priority: first, by *completion percentage*, from high to low; second, by the decision computed by the ARACHNE validation tool [24], from failed to pass; third, by the confidence score computed by ARACHNE, from low to high; and then last, by the cost, from low to high. The *completion percentage* is the percentage of the AC's evidence nodes that are supported by evidence, i.e., have non-missing evidence. This interface is designed to target an assurance-driven development where risk and cost trade-offs are performed in the early stages of the product life cycle, i.e., during discovery, design, and implementation, with the purpose to optimize the development process for assurance. In the early stages, unlike certification, a variety of different assurance case arguments and evidence options can be considered and weighted for their risk reduction benefit and cost.

By double clicking on a candidate in the top ten table of the case summary dashboard, the EVAL tool brings up a dashboard view of the certification argument. This view, shown in Figure 2, is referred to as the candidate dashboard. In the candidate dashboard, clicking on the pie charts brings up another view with more details. For example, clicking on the available artifact chart brings up a table of the available evidence as shown in Figure 3. There is a menu on the left side that can show tabular representation of *Evidences*, *Defeaters*, and *Atomic Arguments*. Clicking on *Evidences* reveals the Evidence View, which is the table of all the evidence (available or missing) needed to support the assurance case and the status of the evidence items, including if they are missing, the location of the raw evidence, the decision on the evidence based on confidence evaluation, the model used in the confidence evaluation and the atomic argument using the

evidence. Clicking on the atomic argument brings up the user interface (UI) which shows the corresponding argument. The table of evidence can be sorted by clicking on the column's headers.

EVAL tool contains a combination of graphical, textual, and tabular representations of assurance cases to provide different perspectives for a certification argument. These representations or views are all accessible from the candidate dashboard and are also accessible from each other. The textual representation is based on the Friendly Assurance Notation (FAN) developed by NASA [25]. Clicking on *Atomic Arguments* on the left side menu of the candidate dashboard brings up the Atomic arguments in the textual representation as shown in Figure 4. The textual representation helps the evaluators better understand the details of the argument. The premises in Figure 4 are clickable and clicking on them leads to the argument or the evidence that support the premise.

The textual representations are also displayed in the *Argument Tree* view as shown in Figure 5. In the Argument Tree view, the left side pane shows the entire AC candidate in a "directory tree" form which helps the user navigate the AC structure. Each goal is a "directory" that could be expanded. By clicking on the small triangle icon to the top left of the goal, the goal is expanded into a set of sub-goals. If a goal is only supported by evidence, then it cannot be expanded further. Selecting the goal in the left pane leads to the display of the textual representation of the corresponding argument in the right side pane.

The *Confidence Tree* view in Figure 6 is similar to the Argument Tree view but gives the user more information about the confidence values associated with the arguments and evidence. Instead of showing the textual representation of the argument, the right side pane shows a tabular result of the confidence analysis on the selected argument (its goal and the sub-goals) including the confidence value, the decision made (pass or non-pass), and the model used, if any, for computing the confidence. As further discussed in Section V-A, the confidence models are clickable, which brings up an interactive interface for the user to probe them.

Graphical representations provide an overview of the argument structure and show the connection between the atomic arguments. They come with different view modes that help identify missing and low confidence artifacts, as well as undeveloped arguments. The graphical representations are referred to as *Argument Structure* view, and is accessible by clicking on corresponding item on the left side menu of the candidate dashboard view shown in Figure 2. This brings up an "upside down tree" like illustration of the AC candidate in which the goals and evidence are the nodes, the root node being the top goal and the leaf nodes the evidence. The edges show the relationships between evidence items and goals. The Argument Structure views are shown in Figures 7 and 8, with the former in a view mode that highlights missing evidence and the latter in the view mode that highlights the confidence levels.

The Argument Structure views provide quick navigation to several other views. As shown in Figure 7, clicking on

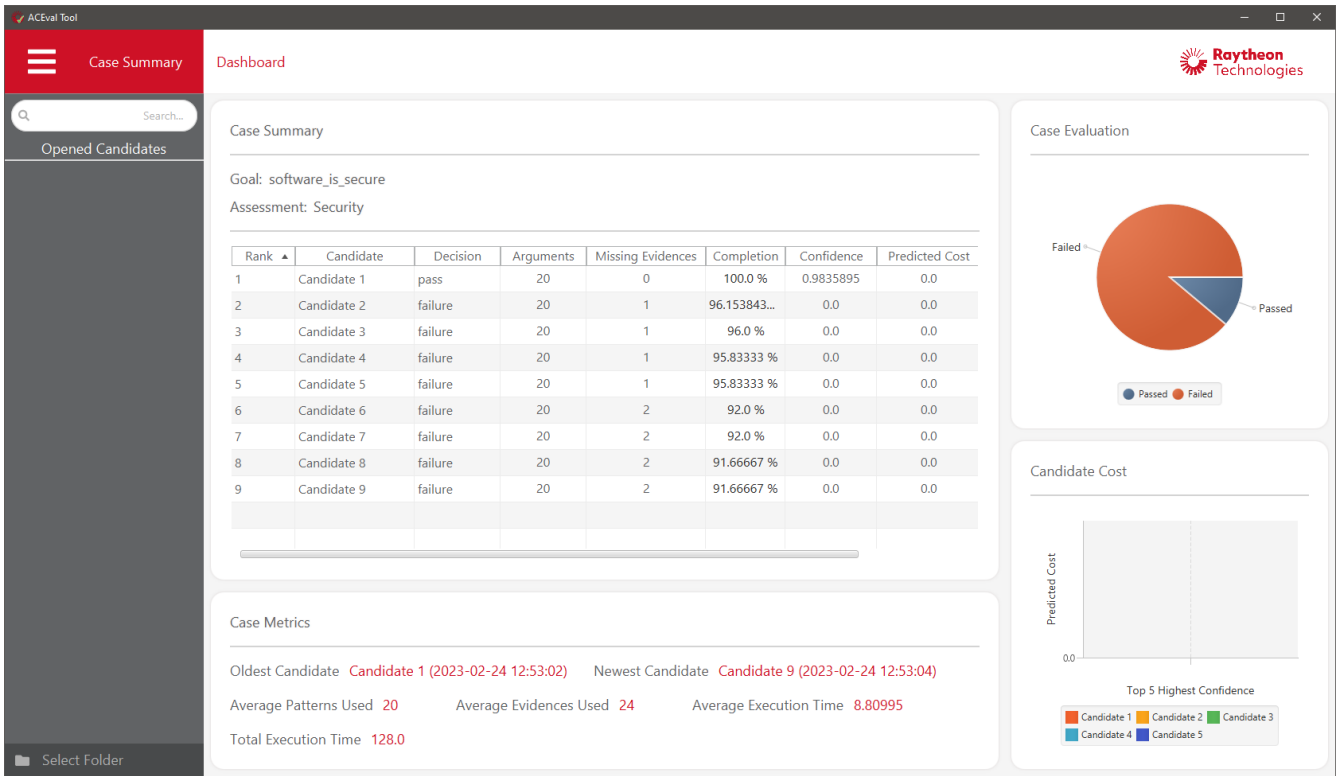


Fig. 1. Case summary dashboard provides a summary of the current top ten certification arguments and various case metrics.

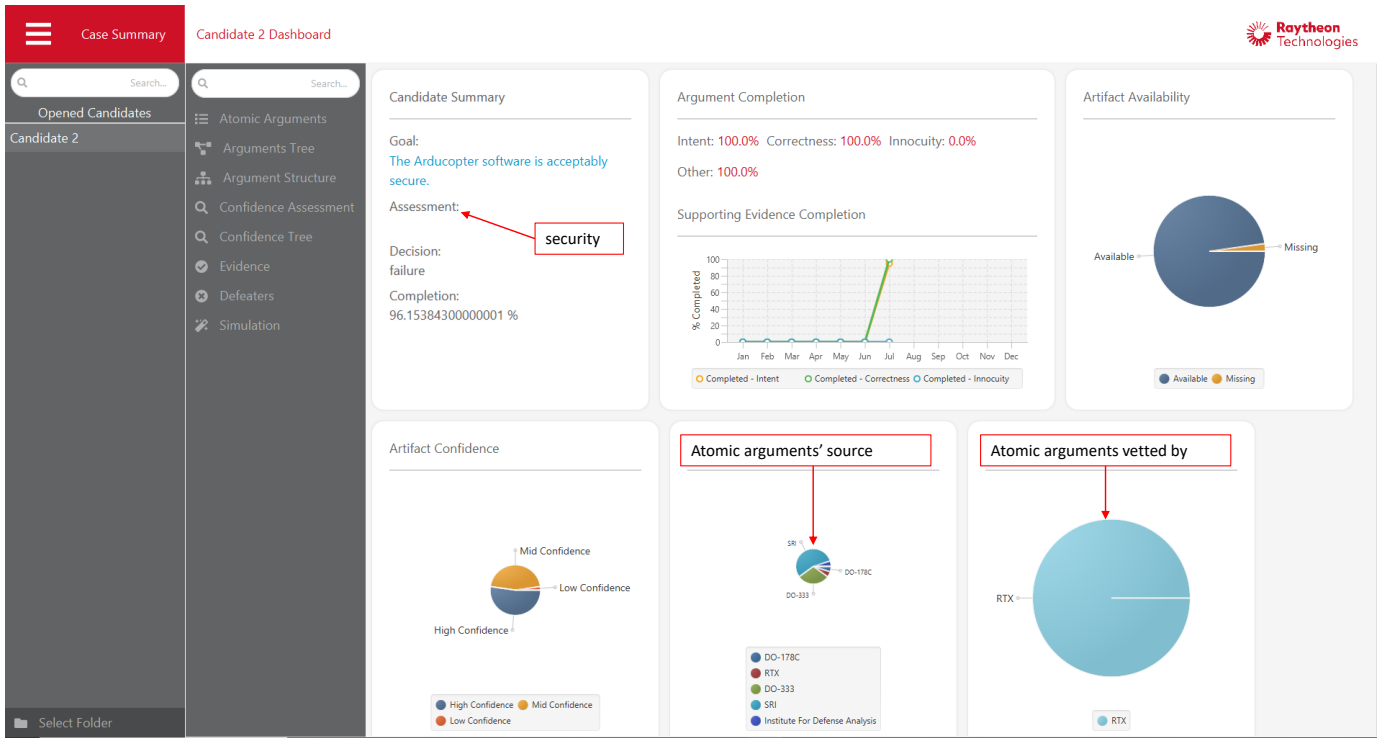


Fig. 2. Candidate dashboard provides a summary of the certification argument that highlights the current progress in the planning and executed argument, sources from the argument patterns, and confidence assessment.

20 Evidences Available









	Evidence	Artifact Location	Missing	Decision	Applied Arguments	C
	CLEAR analysis shows that the requirements satisfy CLEAR_Prop_Req_Accurate.			Passed	requirements_are_correct_consistent_complete	
	CLEAR analysis shows that the requirements satisfy CLEAR_Prop_Req_Consistent.			Passed	requirements_are_correct_consistent_complete	
	CLEAR analysis shows that the requirements satisfy CLEAR_Prop_Req_IP_Complete.			Passed	requirements_are_correct_consistent_complete	
	CLEAR analysis shows that the requirements satisfy CLEAR_Prop_Req_No_Mode_Thrashing.			Passed	requirements_are_correct_consistent_complete	
	CLEAR analysis shows that the requirements satisfy CLEAR_Prop_Req_NonAmiguous.			Passed	requirements_are_correct_consistent_complete	
	CLEAR analysis shows that the requirements satisfy CLEAR_Prop_Req_OP_Complete.			Passed	requirements_are_correct_consistent_complete	
	CLEAR analysis shows that the requirements satisfy CLEAR_Prop_Req_Verifiable.			Passed	requirements_are_correct_consistent_complete	
	A manual review or analysis which					

Fig. 3. Evidence View showing the available evidence.

Believing 

The implementation of Arducopter is good.

Is justified by applying

The implementation consists of the Executable Object Code, which comes from the compilation of the Source Code. The software implementation is good when it satisfies all the specifications defined in Intent. The specifications are satisfied when the implementation passes all tests and the tests have sufficient coverage of the specifications.

Source: SRI
Vetted: RTX

To these premises

- 1 The test cases cover the oracles sufficiently (minimum 70%).
- 2 Test results shows that the binary of Arducopter satisfy all the requirements.
- 3 Test oracles are automatically generated from formal models of requirements.

With Confidence

High, calculated by propagation

Fig. 4. Textual representation of the argument based on NASA FAN.

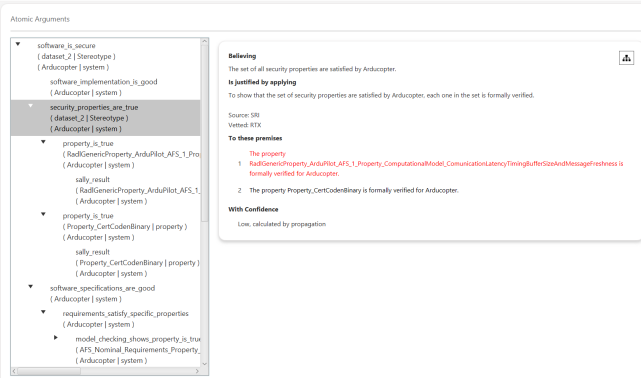


Fig. 5. Argument Tree view shows the textual representation of the arguments, highlighting unsupported premises.

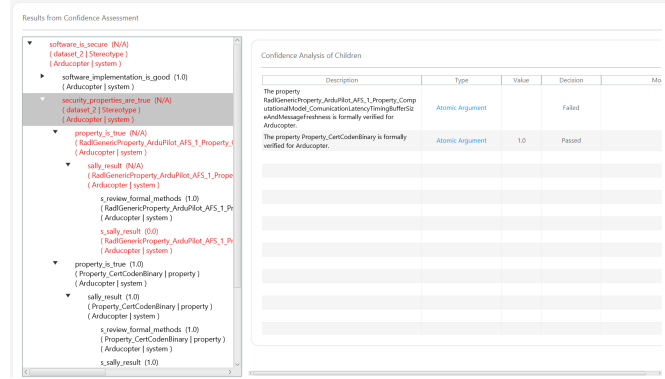


Fig. 6. Confidence Tree view shows the confidence analysis results, with highlighting based on confidence values.

the evidence node (highlighted in red) brings up the missing evidence in the Evidence View as shown in Figure 9. In the undeveloped argument or missing evidence view modes, clicking on an argument node brings up the argument in the Arguments Tree view. On the other hand, in the confidence view mode, clicking on the argument node brings up the argument in the Confidence Tree view. Additionally, as shown in Figure 10, the textual representation of the argument could be displayed in an overlay on top of the Argument Structure view by right clicking on the argument node.

In summary, textual representations aim to facilitate the understanding of the details of the atomic arguments, while graphical representations aim to facilitate the understanding of the entire argument. Both kinds are made available in EVAL with easy navigation between the two. The tabular view, also utilized in the EVAL tool, seems to be the most efficient for summarizing properties that can be sorted, which makes it ideal for showing summaries of evidence, AC candidates, and confidence analysis results.

A. Interactivity

The EVAL tool provides two interactive interfaces. The first one, as shown in Figure 11, provides an interface for users to simulate *what-if* scenarios based on user-defined evidence. Users can interact with check boxes to modify the availability and values of specific evidence items. The validation and assessment of the selected AC can then be re-performed accordingly. This functionality enables users to explore various outcomes of the validation and assessment process, potentially assisting in identifying the generation of new evidence items. For example, if the simulation results indicate that the presence of evidence such as “Model checking shows that GenericProperty_ArduPilot_...” significantly improves the confidence in the analyzed AC, users may decide to generate such evidence based on the simulation.

The second interface, shown in Figure 12, enable users to perform *Sensitivity Analysis* of the models used in the evaluation of confidence. Sensitivity analysis allows users to examine the impact of perturbing the confidence of one claim on the confidence of another claim. For instance, in the

sensitivity analysis, a perturbation in the confidence of the claim “The code and/or binary for the Arducopter component exists” exhibits a minimal influence (0.026) on the confidence of the claim “The property GenericProperty is satisfied”. In contrast, the perturbation in the confidence for the claim “The specifications of Arducopter are correct” has a more substantial effect (0.568). This analysis enhances the interpretability of confidence models, aiding in a more informed elicitation process and providing debugging capabilities.

The simulation and sensitivity analyses are performed by the Automatic Assurance Case Environment (AACE) [26], [27]. The I/O between the interactive features in the EVAL tool and AACE are handled by the AACE *integration framework*, which exposes a RESTful API that enables the EVAL tool to directly run the AACE analyses in the background. The integration framework API communicates the results to the EVAL tool when complete, which are then displayed on the dashboard.

VI. CONCLUSION

We shared the details of a survey on avionics software certification. We presented the distilled perspective on argument-based certification from world-leading certification experts. We detailed the requirements in terms of certification mechanism and rationale, including a discussion of assurance patterns and assurance cases, visualization, documentation or reporting, version control, and feedback collection. We showed ways to facilitate acceptance of a new tool and boost the user confidence in such a tool. The presented EVAL tool provides a sample implementation of a tool for supporting argument-based certification. We expect that the concepts introduced in this paper may resonate with the community and serve as guidelines for the next-generation avionics software certification environments. While the EVAL tool was initially designed for safety certification, we recognize its potential for applications in security certification, especially considering the continuous evolution of threats and security patterns.

REFERENCES

[1] R. Bloomfield and J. Rushby, “Assessing confidence with assurance 2.0,” *arXiv preprint arXiv:2205.04522*, 2023.

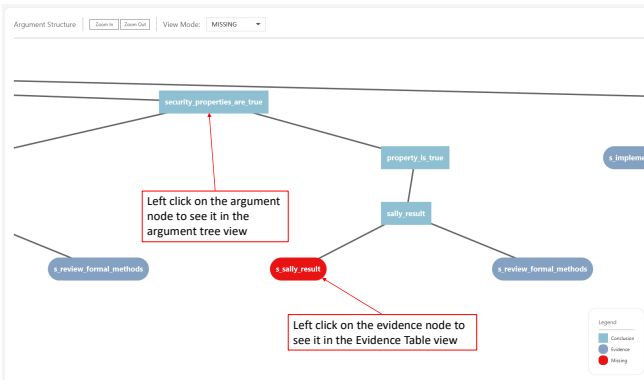


Fig. 7. Structure view with highlighting of missing evidence

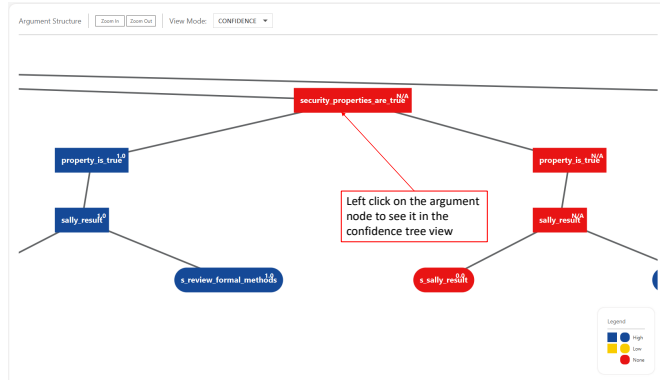


Fig. 8. Structure view with highlighting of confidence levels

Evidence	Artifact Location	Missing	Decision	Confidence Model	Applied Arguments
SALLY analysis shows that AFS_GPS_Location_Property_1 is satisfied by a model of Arducopter.			Passed		sally_result
SALLY analysis shows that AFS_GPS_Communication_Link_Property_1 is satisfied by a model of Arducopter.			Passed		sally_result
SALLY analysis shows that AFS_Instalt_Property_1 is satisfied by a model of Arducopter.			Passed		sally_result
SALLY analysis shows that AFS_Nominal_Requirements_Property_1 is satisfied by a model of Arducopter.			Passed		sally_result
SALLY analysis shows that Property_CertCodenBinary is satisfied by a model of Arducopter.			Passed		sally_result
SALLY analysis shows that RadGenericProperty_ArduPilot_AFS_1_P_Property_ComputationalModel_CommunicationLatencyTimingBufferSizeAndMessageFreshness is satisfied by a model of Arducopter.		x	Failed		sally_result
A review or analysis that shows if properties are satisfied then residual security risk is acceptable.			Passed		security_properties_are_complete
Test oracles are automatically generated from formal models of requirements.			Passed		software_implementation_is_good
Test results shows that the binary of Arducopter satisfy all the requirements.			Passed		software_implementation_is_good
The test cases cover the oracles sufficiently (minimum 70%).			Passed		software_implementation_is_good
A review or analysis that shows if all the specific properties are satisfied then the specifications satisfy the intent of the designers.			Passed		specific_properties_are_complete

Fig. 9. Clicking on the evidence node brings up the particular evidence in the table.

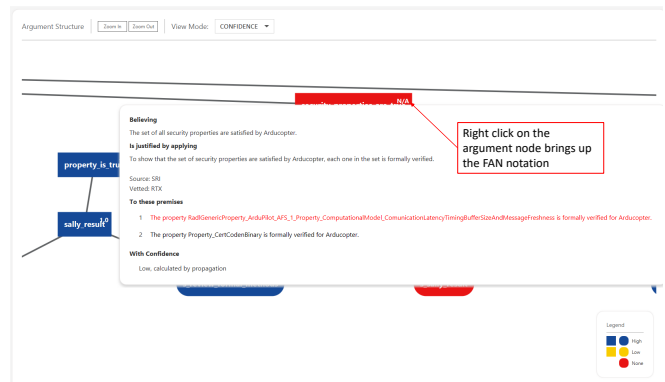


Fig. 10. Right click on the argument node brings up textual representation.

Evidence	Type	Decision	Actual		Simulation		Confidence
			Availability	Value	Availability	Value	
is satisfied by a model of Arducopter.							
SALLY analysis shows that Property_CertCodenBinary is satisfied by a model of Arducopter.	Evidence	Passed	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
SALLY analysis shows that RadGenericProperty_ArduPilot_AFS_1_P_Property_ComputationalModel_CommunicationLatencyTimingBufferSizeAndMessageFreshness is satisfied by a model of Arducopter.	Evidence	Failed	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Test oracles are automatically generated from formal models of requirements.	Evidence	Passed	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Test results shows that the binary of							

Fig. 11. Interactive features: simulation interface.

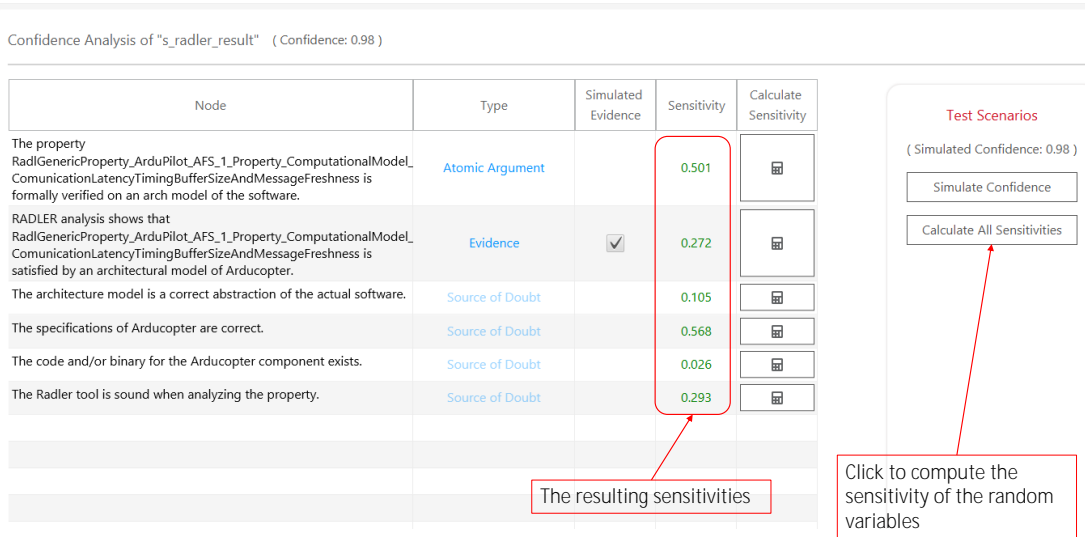


Fig. 12. Sensitivity analysis interface.

- [2] E. Denney and G. Pai, "Tool support for assurance case development," *Automated Software Engg.*, vol. 25, no. 3, p. 435–499, 2018.
- [3] M. Maksimov, S. Kokaly, and M. Chechik, "A survey of tool-supported assurance case assessment techniques," *ACM Computing Surveys (CSUR)*, vol. 52, no. 5, pp. 1–34, 2019.
- [4] A. Moitra, P. Cuddihy, K. Siu, D. Archer, E. Mertens, D. Russell, K. Quick, V. Robert, and B. Meng, "RACK: A semantic model and triplestore for curation of assurance case evidence," in *Computer Safety, Reliability, and Security. SAFECOMP Workshops, 2023*.
- [5] N. Shankar, D. Bhatt, M. Ernst, M. Kim, S. Varadarajan, S. Millstein, J. Navas, J. Biatek, H. Sanchez, A. Murugesan, and H. Ren, "DesCert: Design for certification," *arXiv preprint arXiv:2203.15178*, 2022.
- [6] J. Rushby, "The interpretation and evaluation of assurance cases," Computer Science Laboratory, SRI International, Tech. Rep. SRI-CSL-15-01, 2015.
- [7] R. Bloomfield and J. Rushby, "Assurance 2.0: A manifesto," *arXiv preprint arXiv:2004.10474*, 2021.
- [8] Adelard LLP, *Claims, Arguments and Evidence (CAE)*, 2019, <https://www.adelard.com/asce/choosing-asce/cae.html>.
- [9] The Assurance Case Working Group, "Goal structuring notation community standard (version 3)," 2021.
- [10] R. Hawkins, I. Habli, T. Kelly, and J. McDermid, "Assurance cases and prescriptive software safety certification: A comparative study," *Safety science*, vol. 59, pp. 55–71, 2013.
- [11] L. Sun and T. Kelly, "Safety arguments in aircraft certification," in *4th IET International Conference on Systems Safety. Incorporating the SaRS Annual Conference, 2009*, pp. 1–6.
- [12] E. Denney, G. Pai, and J. Pohl, "AdvOCATE: An assurance case automation toolset," in *International Conference on Computer Safety, Reliability, and Security*. Springer, 2012, pp. 8–21.
- [13] M. R. Barry, "CertWare: A workbench for safety case production and analysis," in *Aerospace conference*. IEEE, 2011, pp. 1–10.
- [14] Y. Matsuno, "D-Case editor: A typed assurance case editor," *University of Tokyo*, 2011.
- [15] Adelard LLP, "Assurance and safety case environment (ASCE)," 2011. [Online]. Available: <https://www.adelard.com/asce/>
- [16] I. Šljivo, B. Gallina, J. Carlson, H. Hansson, and S. Puri, "Tool-supported safety-relevant component reuse: From specification to argumentation," in *Ada-Europe International Conference on Reliable Software Technologies*. Springer, 2018, pp. 19–33.
- [17] I. Šljivo, G. J. Uriagereka, S. Puri, and B. Gallina, "Guiding assurance of architectural design patterns for critical applications," *Journal of Systems Architecture*, vol. 110, p. 101765, 2020.
- [18] J. L. de la Vara, A. Ruiz, and G. Blondelle, "Assurance and certification of cyber-physical systems: The AMASS open source ecosystem," *Journal of Systems and Software*, vol. 171, p. 110812, 2021.
- [19] D. Nešić, M. Nyberg, and B. Gallina, "Product-line assurance cases from contract-based design," *Journal of Systems and Software*, vol. 176, p. 110922, 2021.
- [20] C. Cărlan, V. Nigam, S. Voss, and A. Tsalidis, "ExplicitCase: tool-support for creating and maintaining assurance arguments integrated with system models," in *International Symposium on Software Reliability Engineering Workshops (ISSREW)*. IEEE, 2019, pp. 330–337.
- [21] S. Ramakrishna, C. Hartsell, A. Dubey, P. Pal, and G. Karsai, "A methodology for automating assurance case generation," *arXiv preprint arXiv:2003.05388*, 2020.
- [22] C. M. Holloway, "Understanding the overarching properties," Langley Research Center, National Aeronautics and Space Administration, Tech. Rep. NASA/TM–2019–220292, 2019.
- [23] Z. Daw, S. Beecher, M. Holloway, and M. Graydon, "Overarching properties as means of compliance: An industrial case study," in *IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*. IEEE, 2021, pp. 1–10.
- [24] C. Oh, N. Naik, Z. Daw, T. E. Wang, and P. Nuzzo, "ARACHNE: Automated validation of assurance cases with stochastic contract networks," in *International Conference on Computer Safety, Reliability, and Security*. Springer, 2022, pp. 65–81.
- [25] C. M. Holloway, "The friendly argument notation (FAN)," Langley Research Center, National Aeronautics and Space Administration, Tech. Rep. NASA/TM–2020-5002931, 2020.
- [26] T. Wang, C. Oh, M. Low, I. Amundson, Z. Daw, A. Pinto, M. Chiodo, G. Wang, S. Hasan, R. Melville, and P. Nuzzo, "Computer-aided generation of assurance cases," in *Computer Safety, Reliability, and Security. SAFECOMP Workshops*. Springer, 2023.
- [27] Z. Daw, C. Oh, T. Wang, I. Amundson, A. Pinto, M. Low, M. Chiodo, G. Wang, S. Hasan, R. Melville, and P. Nuzzo, "AAACE: Automated assurance case environment for aerospace certification," in *42nd AIAA/IEEE Digital Avionics Systems Conference*. IEEE, 2023.