From Skepticism to Understanding: Correcting Misconceptions About Overarching Properties

Zamira Daw
Institute of Aircraft System
University of Stuttgart
Stuttgart, Germany
zamira.daw@ils.uni-stuttgart.de

C. Michael Holloway
Safety-Critical Avionics Systems Branch
NASA Langley Research Center
Hampton VA, United States
c.michael.holloway@nasa.gov

***Abstract*—Overarching Properties (OPs) provide a structured framework for defining and evaluating novel means of compliance in software, hardware, and systems, particularly for emerging technologies where approval pathways are still evolving. While OPs are gaining recognition, persistent misconceptions may hinder their broader adoption. This paper systematically addresses five key misconceptions: (1) OPs introduce fundamentally new concepts, (2) OPs are too abstract to be practical, (3) OPs are only applicable to aviation, (4) OPs needs to be compatible with checklist-based compliance, and (5) OPs are synonymous with system properties. By addressing these misconceptions through concrete examples and reference to published case studies, we aim to deepen the understanding of OPs and their role in modern system assurance. This paper contributes to ongoing discussions across industrial, regulatory, and research communities, supporting a more informed and deliberate adoption of OPs.**

***Index Terms*—assurance, argument, Overarching Properties, certification, authorities**

I. Introduction

The original paper describing the Overarching Properties (OPs) [1] introduced the OPs as expressing a sufficient set of properties upon which approval decisions may be made. Three years ago a paper by the second author of this paper rebutted several misconceptions [3] that had arisen regarding the Overarching Properties and the associated concept of Overarching Properties Related Arguments (OPRAs) [2]. This paper extends the previous rebuttal by addressing the following five additional misconceptions:

1) OPs introduce fundamentally new concepts.
2) OPs are too abstract to be practical.
3) OPs are only applicable to aviation.
4) OPs needs to be compatible with checklist-based compliance,
5) OPs are synonymous with system properties.

By addressing these misconceptions, we hope to deepen understanding of the OPs and their potential utility and value in modern system assurance. This paper contributes to ongoing discussions across industrial, regulatory, and research communities, promoting a more informed and deliberate use and adoption of OPs than is possible when misconceptions abound.

Before tackling each of the five misconceptions[1] through concrete examples and referencing published case studies, we provide an abbreviated overview of the history and content of the Overarching Properties. This overview draws heavily from [3], a 2022 DASC paper written by the second author. Some material is used verbatim.

The work on which this paper is based has been sponsored in part by the Federal Aviation Administration through several Interagency Agreements: IAI-1073, Annex 2: Assurance Case Applicability to Digital Systems; IA1-30333, Annex 1: Streamlining Assurance Via Overarching Properties (Savor); and IA1-30333, Annex 2: Using the Overarching Properties in Novel Examples (Opine). Nothing written here, however, may be considered to represent the official views of the FAA, NASA, or the Overarching Properties Working Group (OPWG). Readers should note, however, two of the referenced papers [1], [2] *were* approved by the OPWG before publication.

II. Background

A. History of the OPs

What we now call the Overarching Properties originated in a workshop sponsored by the Federal Aviation Administration (FAA) in December 2005. The FAA selected the invitees to this workshop, seeking to ensure industry and governmental participation from across a wide area of technical disciplines, countries, and assurance viewpoints. The goal of the workshop was to seek to generalize existing assurance objectives for software, hardware, and systems containing both into a small set of what were initially called "meta-objectives." The effort continued with two more invitation-only in-person meetings in April and July 2016 and periodic virtual meetings. These meetings supplemented by conversations in an online forum culminated in a set of three, as they were now more appropriately named, Overarching Properties.

These OPs were presented to the public at the 2016 FAA Streamlining Assurance Processes Workshop, which was held September 13–15, 2016, in Richardson, Texas. Workshop participants expressed opinions across a wide spectrum ranging from deeply skeptical to wildly enthusiastic. Because a large number (and percentage) of opinions were positive, the decision was made to continue the work. Virtual meetings and forum activity continued through the remainder of 2016, resulting in some relatively minor changes to the OPs.

In early 2017 the team was dubbed the Overarching Properties Working Group (OPWG). Most of the work throughout

[1]We are not citing papers or presentations expressing any of these misconceptions. The purpose of this paper is to clarify the misunderstandings or misconceptions, not to condemn or embarrass anyone who holds them.

2017 and 2018 involved trying to develop a collection of criteria for use in evaluating OPs possession, with the intent of publishing a single document describing the OPs and the possession criteria. However, the results of several case studies completed in 2018 identified deficiencies in criteria-based approaches. Consequently, the OPWG decided to publish a description of the Overarching Properties alone. A few minor changes were made during a meeting of the group in April 2019, leading a few months later to the publication of [1].

Continuing work over the next couple of years culminated in the publication of "An Introduction to Constructing and Assessing Overarching Properties Related Arguments (OPRAs)" [2] in January 2022.

B. Details of the OPs

This section provides a rapid introduction to the details of the Overarching Properties as they were described in the defining document. The description is divided into labeled property statements, definitions, requisites, assumptions, and constraints. The meaning of the properties is fully described by the three property statements (not including the labels) and the eight definitions.

Labeled Property Statements

The labels (Intent, Correctness, and Innocuity) are just that: labels. They have no semantic content, having been created solely for convenience of reference. The meaning of the properties would not change if we replaced the labels Intent, Correctness, and Innocuity with the letters A, B, and C; the numbers 1, 14, and 22; the animal names cat, dog, and bird; the proper names Larry, Darryl, and the other property Darryl; or, had we had wanted to be downright unkind, Correctness, Innocuity, and Intent.

Here are the statements as they appear in [1]:

a. **Intent**: The DEFINED INTENDED BEHAVIOR is correct and complete with respect to the DESIRED BEHAVIOR.
b. **Correctness**: The IMPLEMENTATION is correct with respect to its DEFINED INTENDED BEHAVIOR, under FORESEEABLE OPERATING CONDITIONS.
c. **Innocuity**: Any part of the IMPLEMENTATION that is not required by the DEFINED INTENDED BEHAVIOR has no UNACCEPTABLE IMPACT.

The meaning of words and phrases shown LIKE THIS (for example, DESIRED BEHAVIOR and DEFINED INTENDED BEHAVIOR) is given explicitly in a definitions section replicated here:

Definitions

a. DESIRED BEHAVIOR: Needs and constraints expressed by the stakeholders (this includes those needs and constraints identified by the SAFETY ASSESSMENT and those mandated by regulations).
b. DEFINED INTENDED BEHAVIOR: The record of the DESIRED BEHAVIOR.
c. IMPLEMENTATION: ITEM or combination of inter-related ITEMS for which acceptance or approval is being sought.
d. ITEM: a hardware or software element having bounded and well-defined interfaces.
e. FORESEEABLE OPERATING CONDITIONS: External and internal conditions in which the system is used, encompassing all known normal and abnormal conditions.
f. UNACCEPTABLE IMPACT: An impact that compromises the SAFETY ASSESSMENT.
g. SAFETY ASSESSMENT: The systematic identification of FAILURE CONDITIONS and classifications in an operational context, evaluation of the architecture against safety objectives arising from these hazards, evaluation of potential common modes and threats, defining additional intended behaviors to support claims within these evaluations and showing that the safety objectives are satisfied by the IMPLEMENTATION.
h. FAILURE CONDITION: "A condition having an effect on the aircraft and/or its occupants, either direct or consequential, which is caused or contributed to by one or more failures or errors, considering flight phase and relevant adverse operational or environmental conditions or external events." (The OPWG took this language directly from [4].)

The defining document [1] explains each of these definitions in detail. For this paper, we discuss only the first two, which are original phrases coined specifically during the development of the OPs: DESIRED BEHAVIOR and DEFINED INTENDED BEHAVIOR.

DESIRED BEHAVIOR is the term we chose to give to the needs and constraints expressed by the stakeholders. Crucially, these needs and constraints must include those identified by safety assessment and regulations.

DEFINED INTENDED BEHAVIOR is the phrase used for the written down expression of the needs and constraints. The abbreviation DIB is commonly used instead of the full phrase.

Some readers may wonder why we did not use more familiar words such as *requirements* and *specifications*. To paraphrase the discussion of the issue in [1], we chose to not use common but overloaded terms (such as *requirements*, *validation*, or *verification*) to combat the natural tendency of people to subconsciously ignore explicitly provided definitions in favor of relying on their preexisting understanding of the meaning of defined terms instead. We hoped that eschewing ambiguous common terms would increase the likelihood that people would read and rely on the explicit definitions. Our hope has been partially realized, but not to the full extent of the dreams of the most optimistic among us.

The statements and definitions fully define the Overarching Properties. The remaining sections of the description—requisites, assumptions, and constraints—only affect the means by which property possession may be shown.

Requisites enumerate things that must exist to allow the showing of possession of the Overarching Properties:

a. DEFINED INTENDED BEHAVIOR exists.
b. FAILURE CONDITIONS are defined.
c. The record of the SAFETY ASSESSMENT exists.

d. The record of the FORESEEABLE OPERATING CONDITIONS exists.
e. The IMPLEMENTATION exists.
f. Development Assurance Level (DAL) assignments based on FAILURE CONDITION classifications exist.

Assumptions need only be stated, not justified:

a. Stakeholders have the knowledge to express the DESIRED BEHAVIOR.
b. Performing SAFETY ASSESSMENT is not covered by these Overarching Properties.

Constraints directly circumscribe how Overarching Property possession must be demonstrated.

a. The process to ensure possession of the Overarching Properties must be defined and conducted as defined.
b. The means by which the DEFINED INTENDED BEHAVIOR is shown to be correct and complete is commensurate with the DAL.
c. Criteria for evaluating the artifacts are defined and shown to be satisfied individually and collectively.
d. All artifacts are under configuration management and change control.
e. When tiers of decomposition are used, the means of showing correctness among the tiers and to the DEFINED INTENDED BEHAVIOR must be defined and conducted as defined.
f. The IMPLEMENTATION must be correct when functioning as part of the integrated system or in environment(s) representative of the integrated system.
g. All design and manufacturing data to support consistent replication of the type design and instructions for continued airworthiness must be established.
h. The SAFETY ASSESSMENT must address all of the IMPLEMENTATION.

With this background explained, readers without any prior knowledge of the Overarching Properties should be able to read the rest of this paper without getting lost. We discuss in turn each of the misconceptions listed in Section I.

III. Misconception: OPs introduce fundamentally new concepts

We begin with the easiest misconception to correct. As explained in Section II-A the purpose of the effort from the beginning was "to seek to generalize existing assurance objectives." The participants in the initial workshops and the members of the follow-on OPWG stuck firmly to this purpose throughout their work. At no time did they seek to create new concepts or principles for assurance; rather they sought to articulate a new way—a simpler, less domain-specific, less techniques-bound, more flexible way—to express the existing, well-accepted concepts and principles that have contributed to the remarkable safety record of aviation.

The three Overarching Properties are firmly grounded in well-established system development and assurance principles. Current standards and guidelines in aviation and other industries almost uniformally seek to facilitate or assure that any product created to comply with them will be specified properly, do all of the things it is specified to do, and do nothing it is not supposed to do. This trio of necessary characterisitics has been around for a long time. The OPs labeled Intent, Correctness, and Innocuity fully encapsulate each member of the trio. To avoid misconceiving the OPs as introducing fundamentally new concepts, remember this: **The Overarching Properties are a novel expression of well-established principles**.

IV. Misconception: OPs are too abstract to be practical

Some critics argue that OPs are too abstract to be practical or are merely an academic exercise. However, a growing and diverse body of literature demonstrates that this abstraction is not a limitation but a strategic strength. The OPs—Intent, Correctness, and Innocuity—are deliberately high-level to allow flexibility across a range of system architectures, technologies, and development paradigms, including those not covered by traditional certification standards. This generality makes them particularly suitable for emerging technologies like AI and formal methods.

The following case studies illustrate how OPs are being explored as the basis for developing means of compliance across classical software systems, tool qualification, and AI-enabled systems.

A. Systems Using Classical Software

- Micro UAV Demonstrator (μXAV) [5]: A modular UAV demonstrator with multiple subsystems (EPS, HBS, MMS). Used as a testbed to explore how OPs structure assurance arguments in multi-subsystem avionics environments.
- SAFEGUARD [6], [7]: A geofencing system for UAVs where OPs were applied retrospectively to build structured assurance arguments. Demonstrates how legacy systems can still be evaluated effectively under OPs.
- Physical Model for UAV [7]: Uses a physical model as a substitute for traditional requirements to define intended behavior. Demonstrates OPs as a viable alternative means of compliance when standard methods are insufficient.
- Adaptive Trajectory Predictor [8]: Applies adaptive stress testing to validate system behavior under edge cases. OPs guide evidence collection and ensure robustness of behavior under uncertain conditions.
- Autopilot System [9]: Combines natural language and formal methods to show compliance. Natural requirements follow the DO-178C route, while formal specifications are validated against OP-derived properties.
- Auxiliary Power Unit Control System [10]: A traditional aircraft subsystem analyzed using OPs to contrast arguments for DAL A vs. DAL D. Highlights the scalability of OP-based arguments for systems with varying criticality.

B. Tool Qualification

- QGen – TQL-1 Code Generator [11]: An automatic code generator for Simulink/Stateflow. OPs enable tailoring the qualification argument based on a tool-specific risk assessment, moving away from rigid DO-330 checklists.
- Emmtrix Parallelization Studio (ePS): [12] A parallelizing compiler for multicore embedded systems. OPs were used to structure arguments based on STPA-derived risks, emphasizing Intent (clear purpose), Correctness (verified transformation), and Innocuity (absence of side effects).
- qDDSGen [13]: A minimal, qualifiable code generator for Data Distribution Service (DDS) middleware. OPs are used to justify the deterministic and reusable nature of generated code, reducing manual review and accelerating certification.

C. OPs for AI-Enabled Systems

- Recorder Independent Power Supply System (RIPS) State of Health Subcomponent: [14], [15] An ANN-based battery health monitor within an aircraft power supply system. Although Intent and Correctness could not be fully demonstrated due to data representativeness limitations, Innocuity was established. This outcome highlights the diagnostic power of OPs in identifying assurance gaps in ML systems.
- Automated Peripheral Detection System (ADIMA) [16]: A neural network-based peripheral detection system. OPs revealed challenges in specifying the operational design domain (ODD) and verification coverage, which were left implicit in other approaches.
- Runway Landing Guidance System (RLGS): (Späth et al., 2025) [17] Uses vision-based ML to guide aircraft landings. OPs were used to frame arguments around training data quality, performance metrics, and out-of-distribution detection—highlighting the importance of robust dataset curation.
- Airborne Collision Avoidance System (ACAS): (Späth et al., 2025) [17] A neural network mimicking a validated logic table. OPs structured a hybrid assurance case using both formal verification and runtime safety nets, showcasing compatibility with mixed-methods assurance.
- Engine Condition Monitoring System (ECMS): (Späth et al., 2025) [17] A physics-informed ML system that monitors aircraft engine health using digital twins. OPs helped align evidence from transfer learning and model behavior with system-level safety goals.

D. Discussion

The case studies involving systems using classical software provide a familiar foundation for understanding how OPs can be applied in practice. These examples show that OPs can support novel means of compliance, allowing applicants to take credit for assurance activities that enhance safety or reliability—even when those activities are not explicitly recognized by existing certification standards. Importantly, they also demonstrate the applicability and scalability of OPs to real-world, safety-critical systems across different assurance levels.

In the tool qualification case studies, OPs are used to construct tailored and tool-specific assurance arguments. This flexibility enables more efficient qualification processes, especially for modern development tools like code generators and compilers. By grounding assurance in clearly defined OP-based arguments, these studies highlight how OPs can help streamline certification efforts and reduce qualification costs, while maintaining rigor and traceability.

Finally, the case studies focused on AI-enabled systems underscore the unique value of OPs in a rapidly evolving technological landscape. OPs help to identify critical limitations in current practices—such as inadequate data assumptions, lack of model robustness, insufficient traceability, and challenges in verification. For a technology like AI, where traditional standards fall short, OPs offer the community a much-needed common language and evaluative framework to assess the completeness and credibility of assurance efforts at the system level.

To avoid misconceiving the OPs as being too abstract to be practical, remember this: **Real and realistic projects from multiple domains are using the OPs successfully today**.

V. Misconception: OPs are only applicable to aviation

Contrary to this belief, OPs are not limited to the aviation domain. To challenge this misconception, this section first identifies the specific elements of the OPs definition that are tied to aviation and then examines how the core assurance principles of OPs—Intent, Correctness, and Innocuity—are reflected in other industry standards. Although OPs were originally developed by experts in the aviation field, only a small portion of the official description is explicitly aviation-specific, as outlined below:

- The definition for FAILURE CONDITION: "A condition having an effect on the aircraft and/or its occupants, either direct or consequential, which is caused or contributed to by one or more failures or errors, considering flight phase and relevant adverse operational or environmental conditions or external events."
- Requisite f: Development Assurance Level (DAL) assignments based on FAILURE CONDITION classifications exist.
- Constraint b: The means by which the DEFINED INTENDED BEHAVIOR is shown to be correct and complete is commensurate with the DAL.
- Constraint g: All design and manufacturing data to support consistent replication of the type design and instructions for continued airworthiness must be established.

Tailoring these bits as appropriate for a different domain is not difficult. Everything else is either clearly domain agnostic or easily interpreted within the appropriate domain contexts. The following subsections show the alignment of the core concepts of other domains to the OPs.

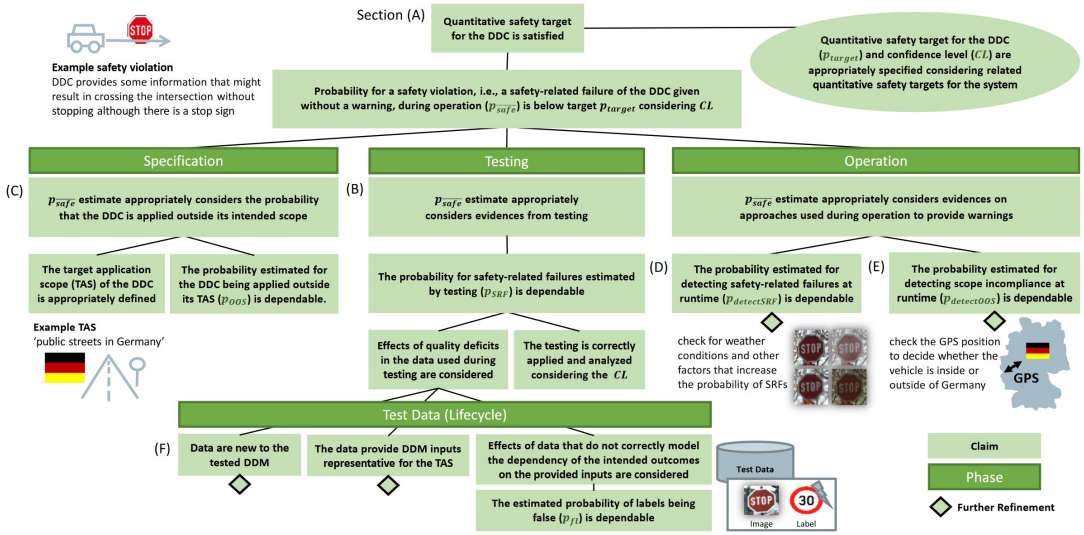

Fig. 1. Assurance case structure for an automotive application proposed in [18]

A. Automotive (ISO 26262, SOTIF)

In the automotive domain, standards such as ISO 26262 and ISO/PAS 21448 (SOTIF) already reflect assurance concepts that align closely with OPs:

- Intent-related: ISO 26262 requires the definition of safety goals and functional safety requirements, which establish the intended system behavior under both normal and faulty conditions.
- Correctness-related: ISO 26262 mandates detailed verification and validation activities across development phases (e.g., software unit verification, integration testing) to demonstrate that the system satisfies its requirements.
- Innocuity-related: SOTIF addresses risks arising from insufficient specifications or limitations in the operating environment — focusing explicitly on unintended behaviors even when the system operates correctly within its design limits.

Thus, the OP structure—comprising Intent, Correctness, and Innocuity—naturally aligns with the assurance processes already established in automotive safety engineering. A specific example is presented in [18], illustrated in Figure 1. This example clearly demonstrates the structural similarity to OPs:

- Intent-related: The system begins by defining a quantitative safety target for the Data-Driven Component (DDC), thereby setting a clear criterion for what constitutes ”safe behavior” in relation to system-level objectives.
- Correctness-related: Assurance is supported through rigorous testing activities, where the analysis focuses explicitly on data quality, correct application of test methods, and appropriate consideration of statistical confidence levels. These practices ensure that the DDC behaves as intended under expected conditions.
- Innocuity-related: During operation, monitoring mechanisms are deployed to detect safety-related failures and scope violations at runtime. These measures are designed to capture cases where the system operates outside its intended domain or encounters unforeseen conditions, thereby addressing unintended behavior and ensuring continued operational safety.

B. Railway (CENELEC EN50128 and EN50129)

In the railway sector, standards such as CENELEC EN 50128 and EN 50129 emphasize a structured assurance process:

- Intent-related: Safety requirements are derived systematically from hazard and risk analyses, captured in documents like the System Requirements Specification (SRS), and traced through to software requirements and design. This ensures the system does the right thing in all defined operational contexts.
- Correctness-related: Detailed verification and validation (V&V) activities are mandated at each lifecycle stage—from software architecture to source code—to demonstrate that the implementation faithfully realizes the intended functionality. High-integrity levels (SIL 3–4) require independent verification and often formal methods.
- Innocuity-related: The standards require systems to be demonstrably safe not only under expected operation but also in the presence of faults, failures, and misuse. This includes fail-safe design principles, error detection mechanisms, defensive coding practices, and exhaustive hazard coverage in the safety case.

C. Medical Devices (ISO 13485, ISO 14971, FDA 21 CFR Part 820)

In the medical device sector, regulatory frameworks such as ISO 13485, ISO 14971, and the FDA’s 21 CFR Part 820 establish a structured, process-oriented quality management system (QMS) focused on product safety and regulatory

compliance. These frameworks naturally align with the core assurance principles of OPs:

- Intent-related: Medical device standards require that the intended use, user needs, and regulatory expectations be explicitly defined at the outset. ISO 13485 mandates a quality management manual, the definition of key procedures, and detailed design and development planning. These must be traceable from initial requirements through to final product realization.
- Correctness-related: ISO 13485 requires that design outputs be verified against design inputs and validated against user needs. Validation is especially critical for processes whose results cannot be fully verified post-production—such as sterilization or software—ensuring correctness even in opaque or high-risk contexts. The standard also enforces traceability of product components and production steps, linking implementation back to its defined intent.
- Innocuity-related: Risk management, as defined in ISO 14971, addresses unintended behaviors and residual risks. It encompasses hazard identification, risk estimation, control measures, and residual risk evaluation, as well as post-market surveillance. ISO 13485 further requires ongoing monitoring through complaint handling, adverse event reporting, and communication with regulatory authorities—ensuring no part of the implementation causes unacceptable impact.

In summary, while a few OP constraints reference aviation-specific terminology, the underlying principles—Intent, Correctness, and Innocuity—are domain-agnostic and readily adaptable to other regulated industries. In fact, these foundational principles are already embedded in the assurance frameworks of other safety-critical domains such as automotive, medical devices, and railway systems.To avoid misconceiving the OPs as being aviation specific, remember this: **Tailoring the OPs for non-aviation domains is not hard**.

VI. Misconception: OPs are fully compatible with checklist-based compliance

Today, certification standards typically define clear objectives that must be achieved to demonstrate system safety. These objectives are based on broad consensus among industry, academia, and regulators and are embedded in the standards themselves.

At the start of the certification process, applicants negotiate with authorities on the specific activities they will undertake to meet the given objectives. This stage focuses on reaching agreement on how the work will be done—not on what the goal is. Once these plans are accepted, the applicant must later demonstrate that the agreed-upon activities have been successfully executed. This model can be seen as a checklist-based approach—where meeting a pre-agreed set of objectives through documented activities forms the basis of compliance. These checklist approaches work well when there is clear agreement on both the objectives and the means to achieve them, which is usually the case in mature domains with stable technologies.

The term "checklist" here is used broadly. It refers not only to literal checklists but also to any compliance method based on fulfilling predefined goals. These methods are efficient and effective in well-understood contexts with minimal uncertainty or ambiguity.

However, the situation changes when we move into the domain of OPs. OPs apply to complex or emerging systems—contexts where there is no broad agreement yet on what the safety objectives should be. In such cases, checklist-style compliance is no longer sufficient. Instead, applicants must also propose and justify the objectives themselves, providing arguments for why their system satisfies the OPs. Thus, the negotiation now extends beyond the "how" (the activities) to also include the "what" (the objectives).

This marks a fundamental shift in certification philosophy: from demonstrating conformance with known standards to reasoning about the safety of novel systems in a structured, evidence-driven way. It is a move from box-ticking toward argued assurance.

This transition has significant implications. It challenges established industry cultures, requires new regulatory processes, and demands that applicants engage more deeply with the reasoning behind safety claims. It is a necessary evolution to ensure safety in technologies for which traditional consensus—and the associated checklist—is not yet available.

To avoid misconceiving the OPs as compatible with a checklist frame of mind, remember this: **Assessing OP possession requires using assessing arguments**.

VII. Misconception: OPs are synonymous with system properties

Another prevalent misconception is that OPs are equivalent to system-specific properties such as explainability, robustness, or reliability. Although OPs are indeed defined as properties that the system must possess to achieve its intended operational goals, the relationship is more nuanced.

Although OPs refer to properties of the system, they are not limited to properties demonstrated directly by the system's behavior. OPs can also be demonstrated through a structured development process that ensures the system will possess the necessary properties, even if this is not directly shown through system-level evidence. In fact, this mirrors current practices in many standards, where specific development processes are prescribed because experience has shown they are sufficient to ensure critical properties without needing to exhaustively prove them at the system level. For example, correctness can be demonstrated either by showing that the final implementation meets its specifications or by following a development process, including verification and analysis methods, that systematically ensures correctness through each development phase.

There have been attempts to extract overarching properties from other domains, such as security or machine learning. In

the case of security, there are well-established system properties like availability, integrity, and confidentiality. While these properties are used to evaluate aspects of a secure system, they are not OPs themselves. Since they define properties of the system that are not, by themselves, sufficient to demonstrate that the system behaves as intended, these properties can instead be integrated into the OP framework by capturing them either in the Intent through specific requirements or by considering their safety impact within Innocuity. For example, availability can be defined in the requirements (Intent), verified through development and testing activities (Correctness), and monitored to detect unintended behaviors that may invalidate them (Innocuity).

This misconception becomes more explicit when people attempt to specify means of compliance using OPs by directly mapping them to machine learning properties such as robustness, generalization, or explainability. Although these properties are relevant for ensuring that the system behaves as intended, they are not sufficient. For instance, robustness can and should be defined in the system requirements, making it part of Intent. However, the specific methods used to ensure the system is robust—such as testing strategies, adversarial testing, or robustness analysis—belong to the demonstration of Correctness. Additionally, there must be a justified argumentation explaining why the chosen methods are sufficient to guarantee the robustness requirements are met. The case of generalization is similar. The degree of generalization required (e.g., the expected performance on unseen inputs) must be captured as part of Intent through explicit requirements. How generalization is validated—such as through the use of test datasets or performance benchmarks—falls under Correctness. Furthermore, given that in many cases the input domain is not bounded, an assurance case must reason about how the system remains safe even when encountering inputs outside the tested bounds, addressing this within the domain of Innocuity.

Another common misuse of OPs is their premature application in automating assurance cases by formally definging OPs using mathematical constructs. While automation of the creation of assurance cases may be a promising idea after a specific means of compliance is well-established, the Overarching Properties and arguments about them are primarily intended to support technologies where the acceptable means of compliance are still unclear.

In these contexts, a manual approach is crucial. Initially, applicants must develop a deep understanding of what constitutes sufficient reasoning, what methods or combinations of methods are necessary, and how these methods satisfy specific assurance goals. This phase also plays a critical role in enabling reviewers and regulatory authorities to identify potential defeaters——gaps, weaknesses, or counterarguments——in the assurance arguments. Thus, the early focus should be on negotiating the means of compliance in a structured, transparent way rather than rushing to automate the assurance process.

Once acceptable means of compliance using the OPs are clearly understood and validated through experience, automation may then play a helpful role. It may help systematically collect relevant artifacts, suggest plausible argumentation schemes, and otherwise support efficient development of a complete assurance case. However, at that stage, the technology is no longer "incoming"; it has transitioned toward maturity. Therefore, while OPs may eventually benefit from automation support, their initial use must remain manual to ensure that the foundational reasoning is sound and that compliance expectations are appropriately negotiated.

To avoid misconceiving the OPs as synonomous with system properties, remember this: **The OPs are more nuanced than well-known system properties such as robustness, dependablity, and the like**.

VIII. Concluding Remarks

Misconceptions are bad. We listed some. Correcting misconceptions is good. We did it. Here is how you can do it, too:

- To avoid misconceiving the OPs as introducing fundamentally new concepts, remember this: **The Overarching Properties are a novel expression of time-honored principles**.
- To avoid misconceiving the OPs as being too abstract to be practical, remember this: **Real and realistic projects from multiple domains are using the OPs successfully today**.
- To avoid misconceiving the OPs as being aviation specific, remember this: **Tailoring the OPs for non-aviation domains is not hard**.
- To avoid misconceiving the OPs as compatible with a check-list frame of mind, remember this: **Showing OP possession requires using arguments**.
- To avoid misconceiving the OPs as synonymous with system properties, remember this: **The OPs are more nuanced than well-known system properties such as robustness, dependability, and the like**.

Acknowledgments

Michael thanks the following people and cats for their contributions to either the work described in this paper, or keeping him (mostly) sane, or both: Annette, Chewie, Darren, George, Kelly, Kim, Leia, Mallory, Mary, Mrs C., Sarah, Sharon, and Srini.

References

[1] C. M. Holloway, "Understanding the overarching properties," Technical Memorandum NASA/TM-2019-219650, National Aeronautics and Space Administration, Hampton, VA, USA, July 2019.

[2] K. S. Wasson and M. Holloway, "An introduction to constructing and assessing overarching properties related arguments (opras)," white paper, NASA Langley Research Center, January 2022.

[3] C. M. Holloway, "False beliefs about the overarching properties and overarching properties relaged arguments," in 2022 IEEE/AIAA 4st Digital Avionics Systems Conference (DASC), September 2022.

[4] European Aviation Safety Agency, "AMC 20-115C software considerations for certification of airborne systems and equipment," available at http://easa.europa.eu/system/files/dfu/annex

[5] J. Chelini, J. L. Camus, C. Comar, D. Brown, A.-P. Porte, M. de Almeida, and H. Delseny, "Avionics Certification: Back to Fundamentals with Overarching Properties," in 9th European Congress on Embedded Real Time Software and Systems (ERTS 2018), 9th European Congress on Embedded Real Time Software and Systems (ERTS 2018), (Toulouse, France), Jan. 2018.

[6] M. Graydon, "Retrospectively Documenting SAFEGUARD's Possession of the Overarching Properties," in 2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks – Supplemental Volume (DSN-S), pp. 27–28, June 2019.

[7] Z. Daw, S. Beecher, M. Holloway, and M. Graydon, "Overarching Properties as means of compliance: An industrial case study," in 2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC), pp. 1–10, Oct. 2021. ISSN: 2155-7209.

[8] M. Durling, H. Herencia-Zapana, B. Meng, M. Meiners, J. Hochwarth, N. Visser, R. Lee, R. Moss, and V. T. Valapil, "Certification Considerations for Adaptive Stress Testing of Airborne Software," in 2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC), pp. 1–9, Oct. 2021. ISSN: 2155-7209.

[9] T. E. Wang, Z. Daw, P. Nuzzo, and A. Pinto, "Hierarchical Contract-Based Synthesis for Assurance Cases," in NASA Formal Methods (J. V. Deshmukh, K. Havelund, and I. Perez, eds.), (Cham), pp. 175–192, Springer International Publishing, 2022.

[10] Z. Daw, S. Beecher, and M. Holloway, "Leveling Arguments: Easier Said Than Done," in 2023 IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC), pp. 1–10, Oct. 2023. ISSN: 2155-7209.

[11] M. A. Aiello, C. Comar, and J. F. Ruiz, "An assurance case based on overarching properties for a tql1 code generator," ERTS2020, 2020.

[12] M. Ibrahim, U. Durak, A. Ahlbrecht, O. Oey, and T. Stripf, "Chasing the Rainbow: Streamlined Tool Qualification," in AIAA SCITECH 2023 Forum, (National Harbor, MD & Online), American Institute of Aeronautics and Astronautics, Jan. 2023.

[13] P. Pazandak, F. Bertocci, B. Razet, and B. Senese, "Fielding Faster: Removing Time and Cost Barriers to Software Certification Using Qualifiable Code Generators," in 2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference (DASC), pp. 1–8, Sept. 2024. ISSN: 2155-7209.

[14] S. Paul, D. Prince, N. Iyer, M. Durling, M. Meiners, L. Tang, B. Meng, N. Visnevski, U. Mandal, and W. Schnepp, "Overarching properties for the assurance of ai/ml-based aerospace systems: Application on a dal d use case phase 2. final report," tech. rep., United States. Department of Transportation. Federal Aviation Administration, February 2023.

[15] S. Paul, N. Iyer, D. Prince, L. Tang, M. Durling, M. Meiners, B. Meng, N. Visnevski, and U. Mandal, "Assurance of AI/ML-Based Aerospace Systems Using Overarching Properties," in 2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference (DASC), pp. 1–10, Sept. 2024. ISSN: 2155-7209.

[16] B. Luettig, Y. Akhiat, and Z. Daw, "ML meets aerospace: challenges of certifying airborne AI," Frontiers in Aerospace Engineering, vol. 3, p. 1475139, 2024. Publisher: Frontiers Media SA.

[17] H. Späth, T. Varchev, S. Staudacher, Z. Daw, and M. Holloway, "Arguing machine learning assurance for certification," p. 10.18420/se2025, Gesellschaft für Informatik, Bonn, 2025. ISSN: 2944-7682.

[18] M. Kläs, L. Jöckel, R. Adler, and J. Reich, "Integrating Testing and Operation-related Quantitative Evidences in Assurance Cases to Argue Safety of Data-Driven AI/ML Components," Feb. 2022. arXiv:2202.05313.