Adapting ISO/PAS 8800 to AI and ML System Safety Assurance within Other Industries

Ehsan Ghahremani – Critical Systems Labs Inc.
Jonathan Groves – Critical Systems Labs Inc.
Jeff Joyce – Critical Systems Labs Inc.
Laure Millet – Critical Systems Labs Inc.

WWW.CRITICALSYSTEMSLABS.COM

Adapting ISO/PAS 8800 to AI and ML System Safety Assurance within Other Industries

Ehsan Ghahremani
Critical Systems Labs Inc.
Vancouver, Canada
ehsan.ghahremani@cslabs.com

Jonathan Groves
Critical Systems Labs Inc.
Vancouver, Canada
jonathan.groves@cslabs.com

Jeffrey Joyce
Critical Systems Labs Inc.
Vancouver, Canada
jeff.joyce@cslabs.com

Laure Millet
Critical Systems Labs Inc.
Vancouver, Canada
laure.millet@cslabs.com

***Abstract* — Most commonly used safety standards were developed before the widespread adoption of Artificial Intelligence (AI) and Machine Learning (ML) systems, and therefore do not address the safety implications of AI/ML systems and components. As AI/ML use expands, particularly in safety-critical systems, a framework for their safe use is urgently needed.**

Across industries, current safety standards derived from IEC 61508, such as ISO 26262 (automotive), EN 50128 (rail), IEC 61511 (process), and other sector-specific standards such as DO-178C (aerospace) provide no explicit guidance on AI/ML. These will be referred to in this paper as "conventional" standards.

ISO/PAS 8800, a recent automotive-focused standard for AI safety, offers explicit guidance for AI/ML systems. This whitepaper by Critical Systems Labs (CSL) generalizes ISO/PAS 8800's key recommendations to other industries, highlighting how ISO/PAS 8800 can supplement conventional standards in addressing AI/ML safety.

Keywords —Artificial Intelligence (AI), Machine Learning (ML), Functional Safety, Safety Assurance Case, Structured Argumentation, IEC 61508, ISO 26262 (automotive), EN 50128 (rail), IEC 61511 (process), DO-178C (aerospace).

I. INTRODUCTION

High-integrity systems across industries are increasingly incorporating Artificial Intelligence and Machine Learning (AI/ML) into their system. Conventional (i.e., non-AI/ML) safety standards provide no guidance on how AI/ML should be addressed within safety-critical designs. This creates a significant gap.

ISO/PAS 8800 was developed to supplement ISO 26262 for automotive systems, specifically with AI/ML in mind. Critical Systems Labs (CSL) believes that its guidance is equally relevant to other sectors, including aerospace, rail, process industries, and others.

For instance, both IEC 61508 and EN 50128 (rail) mention AI only in the context of fault correction. EN 50128 Annex D.1 states that "Artificial Intelligence Fault Correction" may support forecasting, correction, maintenance, and supervisory actions when rules are derived directly from specifications [1]. This reflects an earlier, deterministic view of AI, not today's probabilistic ML models. In their latest versions (IEC 61508:2010 and EN 50128:2020), neither standard addresses probabilistic AI/ML approaches, instead limiting "AI" to specification-driven deterministic systems such as lookup tables.

Similarly, ISO 26262 (automotive), IEC 61511 (process), and DO-178C (aerospace) contain no references to AI or ML systems. DO-178C, for example, assumes deterministic software, emphasizing requirements-based testing, structural coverage, partitioning, and configuration management, but not probabilistic or data-driven AI/ML behavior.

Adapting ISO/PAS 8800's provisions can provide an interim framework for AI/ML assurance until AI/ML-specific standards are created. IEC 61508 allows for justified customization of the software safety lifecycle, for example, by stating: *"Any customization of the software safety lifecycle shall be justified on the basis of functional safety."* CSL views ISO/PAS 8800 as a valid reference for such customization in AI/ML contexts. This applies across lifecycle phases, including:

- Evaluating AI/ML use in design,
- Selecting training data,
- Defining requirements,
- Mitigating AI/ML-specific risks,
- Building assurance cases, and
- Managing lifecycle impacts.

Overall, this approach is intended to create a robust argument for the use of AI/ML and that safety considerations are adequately considered. As per IEC 61508, "*The specification of the requirements for safety-related software shall be sufficiently detailed to allow the design and implementation to achieve the required safety integrity …, and to allow an assessment of functional safety to be carried out*."

The practical takeaway is clear: organizations should understand the specific modifications required when AI/ML is introduced and employ mitigation strategies that make those mitigations actionable.

II. AI/ML SAFETY ASSURANCE CASE(S)

A major addition in ISO/PAS 8800 is the explicit requirement for an AI/ML safety assurance case. Such cases must be comprehensive, at a system-level, and kept current as either the AI/ML system or its environment changes.

ISO/PAS 8800 emphasizes the importance of safety assurance cases for AI/ML systems and outlines structured approaches in its sections on "Assurance arguments for AI systems" and "Safety analysis of AI systems" [2]. These are not merely optional guidance; instead, they are critical tools for coordinating safety analysis, organizing testing activities, and assembling evidence to demonstrate that risk posed by AI/ML systems is acceptably mitigated. A structured safety

assurance case not only organizes evidence into a coherent, accessible form but also captures the logical reasoning that supports (or challenges) the adequacy of that evidence. This makes it easier to identify and address risks or design gaps that could otherwise go unnoticed. Smaller, component-level assurance cases can be created for individual AI/ML elements when a "bottom-up" approach to system safety is taken.

Building AI assurance case(s) are expected to complement or be in lieu of other safety analysis techniques such as FTA, FMEA, STPA, ETA, Bayesian networks, or HAZOP.

Two strategies generally exist for demonstrating that a system using AI/ML is safe: bottom-up and top-down.

- **Bottom-up approach**: AI/ML is treated as one component within a larger system. Assurance is established as usual for the non-AI/ML portions, while additional evidence is provided for AI/ML-specific gaps. This approach is especially effective when retrofitting existing systems with AI/ML, where only limited functionality is allocated to it. The argument focuses on ensuring that the AI/ML component does not degrade overall system safety.
- **Top-down approach**: The entire system is treated as an AI/ML-enabled system. Because conventional standards lack prescriptive detail for AI/ML, ISO/PAS 8800 concepts must be used to supplement them. Unlike the bottom-up approach, this strategy reframes system safety from a new perspective rather than retrofitting piecemeal.

DO-178C reflects a similar concept to our top-down approach in its treatment of software levels: "*If partitioning and independence between software components cannot be demonstrated, the software components should be viewed as a single software component when assigning software levels…*"

Regardless of approach, effective assurance requires structured argumentation, such as Goal-Structuring Notation (GSN) or Eliminative Argumentation (EA). These frameworks capture safety activities, justifications, and evidence in a logical and traceable form, directly linked to lifecycle requirements.

III. AI/ML Safety Assurance Case Approach

It is important that a suitable style of safety assurance case is selected. The safety assurance case should be targeted towards an overall purpose and the expected audience.

For example, one assurance case argument might be to argue that the AI/ML system is at least X times safer than an equivalent non-AI/ML system. Another might be to argue that each of the AI/ML-specific safety requirements are met. Yet another might argue that the AI/ML component meets the intent of each clause within a relevant standard. Each of these examples might be better suited for certain audiences, e.g., internal engineering groups, management, verification and validation (V&V) groups, investors, and/or regulatory authorities.

Additionally, the selection of a top-down vs bottom-up approach, scope (e.g., software and/or hardware; the inclusion of cybersecurity), etc., might also lead to different approaches.

For example, if the intent is to seek approval from a regulatory authority, structuring the assurance case in a way that is familiar to them is expected to be beneficial. Additionally, pre-empting the regulatory authority's possible questions with the addition of "doubts" within the assurance case might also be beneficial to demonstrate confidence that risks have been adequately mitigated.

A suitable alternative strategy might be to argue that each AI/ML safety requirement is fulfilled and/or to argue that the intent of a relevant standard is met.

The use of a Safety Assurance Case to organize evidence (e.g., test reports) into a consistent and logical argument is often effective is often beneficial for seeking approval by regulatory authorities.

CSL provides additional guidance in another recent whitepaper titled "*Towards an ISO/PAS 8800:2024 Compliant Assurance Argument: Assurance Case Development for AI and ML*", which delves deeper into software recommendations for an assurance case, and how software-based tooling can aid in the creation of ISO/PAS 8800 assurance cases. This includes topics such as embedding evidence and Key Performance Indicators (KPIs) within the assurance case, stakeholder involvement and reviews, issue tracking, change management, semi-automated report generation, argument searching and filtering, and others [3].

IV. Key ISO 8800 AI/ML Additions

ISO 8800 addresses additional AI/ML gaps present in regulatory frameworks that were not designed with these technologies in mind. Key topics include the selection of AI/ML training data, the creation of AI/ML-specific statistical-based requirements, mitigations for when AI/ML produces unsafe outputs, the use of a safety assurance case, and management of the AI/ML component's lifecycle.

In more depth, CSL has identified the following six gaps that ISO 8800 provides guidance on mitigating:

1. The outputs of the system may be based on probability (e.g., ML models) rather than being deterministic (i.e., conventional software),
2. The quality and selection of training data play a substantial role in the output of the system,
3. New or modified traceability methods might need to be used,
4. Changes might occur over time to AI/ML components (e.g., dynamic ML models) and/or their operating environment,
5. Failure modes of an AI/ML system differ from those of a non-AI/ML system, and
6. An AI/ML model might influence non-AI/ML components of a system.

The following six sections each discuss specific potential mitigations for the above six gaps respectively. It should be noted that this does not constitute a complete list of risks nor mitigations, and exact risks and mitigations will vary from system to system. These concepts build upon ideas introduced in CSL's earlier work, "*Closing the Gap Between IEC 61511 and the Use of Artificial Intelligence in Plant Safety*" [4], but are extended and reframed to highlight their

broader applicability across industries through the lens of ISO/PAS 8800.

A. Probabilistic Outputs from AI/ML Systems

Unlike conventional software, which typically produces deterministic outputs, AI/ML systems often generate results probabilistically. This means that the same input can, under certain conditions, lead to different outputs, including unexpected ones. Such variability contrasts sharply with the logical rule-based behavior of traditional code-centric systems.

ISO/PAS 8800 addresses this by introducing AI/ML-specific safety requirements. These requirements are framed similarly to conventional requirements but are expressed in probabilistic terms. For instance, an AI system may be required to achieve an uncertainty rate no greater than 1e-6, with statistical confidence demonstrated through significance testing (e.g., p-values). This shift explicitly acknowledges that acceptable safety in AI/ML cannot always be defined in absolute terms but must often be specified probabilistically [1].

This marks a departure from conventional standards. DO-178C, for example, assumes determinism from the system rather than probabilistic outputs, as demonstrated by Section 6.4.2 of the standard requiring "*normal range and robustness test cases*," and Section 6.4.4.2 specifying "*structural coverage analysis*" to ensure complete test coverage [5]. EN 50128 adopts a similar stance, requiring repeatable behavior against specifications, e.g., as shown in Sections 6.1.1 and 6.1.4.5 [1]. ISO 26262 Part 6, Section 6, also mandates explicit software safety requirements but does not account for probabilistic outputs. ISO/PAS 8800 extends these frameworks by explicitly including probabilistic requirements.

Given the difficulty of fully enumerating AI/ML requirements, particularly in high-dimensional spaces, alternative techniques are needed. Metamorphic relations can formalize expected transformations in input-output behavior, while metamorphic testing systematically validates that these relationships hold under varied conditions [3].

Even with probabilistic requirements in place, unsafe outputs may still occur, though at low probability. ISO/PAS 8800 suggests bridging this gap through logical reasoning and safety arguments that demonstrate how unsafe outputs are prevented from propagating into unsafe system states. Mitigation measures include:

- Safety supervisors that detect anomalies (e.g., out-of-bound values, excessively rapid changes, or inconsistencies with other systems), and
- Ensemble methods, where conventional algorithms operate alongside AI/ML components to reduce risk exposure.

Together, these approaches provide a structured way to manage the inherent variability of AI/ML systems.

B. Quality and Selection of Training Data

Probabilistic outputs make the role of training data especially critical. The quality of an AI/ML system's behavior is directly tied to the precision, robustness, and representativeness of its datasets.

ISO/PAS 8800 fills an important gap by requiring systematic dataset management. This includes defining dataset requirements, designing datasets, implementing and verifying datasets, validating representativeness, testing dataset adequacy, and managing datasets throughout the lifecycle [1]. Attributes such as accuracy, completeness, independence, temporality, traceability, and verifiability must all be considered to ensure robust outputs.

Conventional standards give little attention to training data. As per Sections 2.5.1 and 11.22 of DO-178C, the standard addresses only "Parameter Data Items" that are static, e.g., controlled datasets like lookup tables. While the standard requires configuration management, it does not address training data representativeness or completeness [5]. In Sections 6.5.4.14-17, EN 50128 emphasizes traceability but similarly omits dataset governance. ISO/PAS 8800 expands into this unaddressed area, aligning dataset management with safety-critical needs.

By recognizing data as a first-class safety artifact, ISO/PAS 8800 ensures that probabilistic system behavior is not left to chance but is grounded in well-specified and managed training processes.

C. Traceability for AI/ML Systems

If data quality determines the foundation of AI/ML behavior, traceability provides the structure needed to ensure safety arguments remain defensible. Traceability links high-level safety requirements through design, implementation, testing, and maintenance.

Conventional standards already require strong traceability. In Sections 5.5. and 11.21, DO-178C mandates bidirectional links between requirements, design, code, and verification results. ISO 26262, Part 8, Section 13.4.3.5.1, requires test-to-requirement traceability. EN 50128 defines traceability as "*the ability to establish relationships among development products, and mandates it across requirements, design, implementation and testing*" [6]. However, all of these standards stop short of datasets. ISO/PAS 8800 closes this gap by explicitly extending traceability to training data. This ensures that AI/ML safety requirements can be traced backward to the datasets that support them and forward to the evidence proving their satisfaction.

Consider an AI/ML-based safety requirement that requires the AI/ML system to detect and report anomalous vibrations within a mechanical system. If the training dataset lacks similar instances of anomalous vibrations, a traceability review considering AI/ML components should expose the gap, prompting corrective actions and adjustments to the dataset, such as the addition of suitable simulated or real-world anomalous vibrations. By linking dataset requirements directly to safety requirements, ISO/PAS 8800 ensures that critical behaviors are adequately represented and tested.

D. The Dynamic Nature of AI/ML Systems

Conventional safety standards assume systems remain static until deliberately modified. AI/ML challenges this assumption. Models may drift over time due to environmental changes, evolving input distributions, component replacements, or shifts in usage patterns. These dynamics can degrade performance relative to the validated baseline.

ISO/PAS 8800 addresses this by requiring proactive monitoring and lifecycle management. Safety assurance must account not only for initial validation but also for ongoing performance under real-world conditions. Monitoring

mechanisms can detect when operational inputs diverge from training distributions, while continuous reassessment ensures prior safety arguments remain valid [1].

Conventional standards provide partial parallels. As per Sections 7.0 and 12.1, DO-178C mandates safety features to manage erroneous inputs and assumes tightly controlled software [5]. EN 50128 requires that any change or enhancement triggers a safety impact analysis, and, in accordance with Sections 6.6.4.2 and 9.2.4.19, those impacts must be addressed by returning to the appropriate lifecycle phase and reapplying all subsequent processes, effectively treating the system as static until it is formally requalified. Neither standard anticipates naturally evolving AI/ML models. IEC 61508 offers more relevant provisions, requiring operational procedures to be updated based on audits and tests, hazard analyses for modifications, and functional testing under environmental conditions. ISO/PAS 8800 extends these concepts to environments that shift in unforeseen ways, for example, new vehicle types in automotive or new lighting conditions in rail systems.

In short, ISO/PAS 8800 reframes lifecycle assurance to treat change as inherent rather than exceptional in AI/ML systems.

E. AI/ML-Specific Failure Modes

AI/ML systems often have unique failure modes compared to non-AI/ML systems. In conventional (i.e., non-AI/ML) systems, the transformation from input to output is typically governed by explicit, human-readable logic, allowing engineers to clearly see which input conditions produce specific outputs. By contrast, AI/ML systems, particularly those using neural networks, often lack this high level of interpretability for two key reasons. First, their input–output mapping is defined by highly complex, high-dimensional, and potentially non-linear mathematical relationships, making it difficult for a human to mentally trace cause and effect. Second, the learned internal representations do not necessarily correspond to human-understandable concepts, meaning there is no clear human-interpretable link between the input features and the model's decision process. Together, these factors make it harder to predict or explain AI/ML behavior, which can lead to unexpected or non-intuitive failure modes.

At a high level, standard testing methods such as component and integration tests using pass/fail criteria are still broadly applicable [2]. The main difference is that AI/ML system testing will often use alternative testing methods within these high-level testing categories to gain confidence in the system's safety. For conventional software for automotive applications, ISO 26262 provides common verification and testing methods within Part 11, Section 10, including static code analysis, requirements-based tests, interface tests, etc.

Methods for testing AI/ML systems can include a range of different approaches, such as statistical testing, metamorphic testing, K-way combinatorial testing, gradient-based search methods, synthetic test case generation, expert knowledge-based testing, adversarial testing, and robustness testing [2]. For example, DO-178C requires the use of robustness test cases in Section 6.4.2.2 of the standard, although these test cases would need to be expanded for AI/ML-specific systems. In CSL's experience, metamorphic testing, K-way combinatorial testing, and expert knowledge-based testing have proven effective for AI/ML system safety analysis [7] [8].

DO-178C also acknowledges complexity in failure progression, e.g., as per Section 2.3.1 which states that "*a software error may be latent…*" and that "*…the sequence of events that leads from a software error to a failure condition may be complex…*" [5]. These provisions address deterministic failures but not opaque AI/ML failure modes. ISO 8800 extends this coverage to emergent ML-specific failures [2].

In Section 3.1.10, EN 50128 defines failure deterministically as the loss of ability to perform as required and provides deterministic fault/error definitions in Sections 3.1.51-52. While Annex D lists techniques such as formal methods, control flow analysis, and inspections, these are suited to deterministic software. Annex D.1 references AI Fault Correction only in the sense of specification-driven fault forecasting and correction, not probabilistic ML. ISO/PAS 8800 extends these frameworks by explicitly considering emergent AI/ML-specific failure modes.

F. Influence of AI/ML Systems on Non-AI/ML Components

A critical concern when introducing AI/ML is its potential influence on interconnected non-AI/ML (or other AI/ML) components. Because these interactions can create cascading risks, understanding and managing interconnections is essential. A clear system definition, detailing AI/ML functionality interfaces (inputs and outputs), and requirements, provides the foundation for assessing such interactions.

Architectural mitigations, particularly isolation strategies, are central to reducing unintended impacts. Conventional standards already recognize this need. For example, DO-178C explicitly requires partitioning to ensure that *"a partitioned software component should not be allowed to contaminate another partitioned software component's code, input/output (I/O), or data storage areas."* It also emphasizes the importance of monitoring, noting that *"safety monitoring is a means of protecting against specific failure conditions by directly monitoring a function for failures that would result in a failure condition. Monitoring functions may be implemented in hardware, software, or a combination of hardware and software."* These principles align closely with ISO/PAS 8800's architectural recommendations, although DO-178C applies only to deterministic systems.

Confidence in AI/ML integration can be further enhanced through rigorous verification and validation activities, including both virtual and physical testing [1]. Complementary frameworks such as AI Safety Integrity Levels (AI-SIL) from CSL's "*Safety Integrity Levels for Artificial Intelligence*" white paper [5] also provide guidance. AI-SIL links the entropy of inputs and the non-determinism of outputs to the level of assurance rigor required, helping organizations determine how much scrutiny is necessary for AI/ML components. Input entropy considers the complexity of inputs to the system. For example, a system that's only input is a single sensor's reading (e.g., water level in a tank) would be a lower-entropy input, whereas camera data which contains a matrix of pixel intensity values (especially when combined with lidar and radar data) would be a higher-entropy input. Similarly, non-determinism assesses the size

and variability of an AI-based function's output space. For example, a binary classification algorithm that's only output is a true/false value of if a pedestrian is detected by an autonomous vehicle would be lower non-determinism, whereas a planned future route of an autonomous vehicle's trajectory would be a higher non-determinism.

Other industry standards provide parallel guidance with respect to the influence of AI/ML systems on other systems. EN 50128 requires software and hardware integration testing to demonstrate correct interactions, as per Section 7.6.1, and mandates that generic software be rigidly separated from application algorithms [1]. However, it does not anticipate the probabilistic influence of AI/ML models on deterministic components.

IEC 61508 makes the case for explicit separation at the system level: "*In the specification, it should be decided whether a separation of the safety-related systems and non-safety-related systems is possible. Clear specifications should be written for the interfacing of the two parts. A clear separation reduces the effort for testing the safety-related systems.*" This statement is directly applicable when integration of AI/ML systems with non-AI/ML and other systems, underscoring the importance of strong architectural boundaries.

V. Conclusion

This paper has outlined how ISO/PAS 8800 can supplement existing conventional safety standards by providing a practical roadmap that addresses aspects not covered in current conventional standard frameworks. Its purpose has been to identify key considerations that arise with the use of AI/ML systems and to demonstrate how these can be argued in a way that remains defensible to independent assessors.

At a high level, the paper emphasizes the need to define AI/ML-specific safety requirements and to establish guidance for managing issues not considered in conventional standards. This ensures that safety assurance is maintained across the full lifecycle of AI/ML systems from initial design, through development, and into ongoing operation as either the system or its environment evolves.

Additionally, ISO/PAS 8800 requires the addition of an AI/ML safety assurance case. This paper provides a brief introduction into potential structures and topics to consider, when creating such a safety assurance case. This includes introduction into the top-down vs bottom-up approach, the need to tailor the case to the audience, and tips such as the inclusion of arguing doubts pre-emptively and organizing test evidence to provide key stakeholders with confidence that risks have been adequately mitigated and concerns have been adequately addressed.

Other relevant CSL papers were also referenced for additional specific information, such as:

- "Towards an ISO/PAS 8800:2024 Compliant Assurance Argument: Assurance Case Development for AI and ML Systems"
- "Safety Integrity Levels for Artificial Intelligence".

Taken together, these methods offer organizations a strategy for adapting established industry safety standards to AI/ML. While DO-178C, IEC 61508, and ISO 26262 provide partial parallels in areas such as robustness testing, fault tolerance, partitioning, and monitoring, ISO/PAS 8800 extends these principles to address the distinct challenges of probabilistic outputs, dataset quality, dataset traceability, dynamic model evolution and non-deterministic failure modes. Similarly, EN 50128 reinforces lifecycle discipline and traceability but does not address these AI/ML-specific issues. In doing so, ISO/PAS 8800 fills critical gaps and establishes a path for the safe integration of AI/ML into safety-critical domains.

VI. Bibliography

[1] "CELENEC EN 50128: Railway Applications - Communication, Signalling and Processing Systems - Software for Railway Control and protection Systems," CELENEC EN, 2020.

[2] "ISO/PAS 8800: Road Vehicles - Safety and Aritificial Intelligence," ISO, 2024.

[3] E. Ghahremani, J. Groves, J. Joyce and L. Millet, "Towards an ISO/PAS 8800:2024 Compliant Assurance Argument: Assurance Case Development for Artificial Intelligence (AI) and Machine Learning (ML) Systems," 2025.

[4] E. Ghahremani, J. Groves, J. Joyce and L. Millet, "Closing the Gap Between IEC 61511 and the Use of Artificial Intelligence in Plant Safety," 2025.

[5] "DO-178C: Software Considerations in Airborne Systems and Equipment Certification," RCTA Inc., 2011.

[6] "ISO 26262: Road Vehicles Functional Safety," ISO, 2018.

[7] L. Millet, "Specifying Functional Safety Requirements for AI/ML in Terms of Metamorphic Relations," CSL, 2024.

[8] S. Diemert, A. Casey and J. Robertson, "Challenging Autonomy with Combinatorial Testing," in *IEEE International Conference on Software Testing*, 2023.

[9] "IEC 61511: Functional Safety - Safety Instrumented Systems for the Process Industry Sector," IEC, 2016.

[10] "IEC 61508: Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems," IEC, 2010.

[11] S. Diemert, L. Millet, J. Groves and J. Joyce, "Safety Integrity Levels for Artificial Intelligence," in *IEEE SafeComp*, 2023.