PESARO 2026

The Sixteenth International Conference on Performance, Safety and Robustness in Complex Systems and Applications

ISBN: 978-1-68558-395-8

May 24 - 28, 2026

Venice, Italy

PESARO 2026 Editors

Rémy Houssin, Université de Strasbourg - ICube Laboratory, France
Juliette Mattioli, Thales, France

PESARO 2026

Forward

The Sixteenth International Conference on Performance, Safety and Robustness in Complex Systems and Applications (PESARO 2026), held between May 24-28, 2026 in Venice, Italy, continued a series of events dedicated to fundamentals, techniques and experiments to specify, design, and deploy systems and applications under given constraints on performance, safety and robustness.

There is a relation between organizational, design and operational complexity of organization and systems and the degree of robustness and safety under given performance metrics. More complex systems and applications might not be necessarily more profitable, but are less robust. There are trade-offs involved in designing and deploying distributed systems. Some designing technologies have a positive influence on safety and robustness, even operational performance is not optimized. Under constantly changing system infrastructure and user behaviors and needs, there is a challenge in designing complex systems and applications with a required level of performance, safety and robustness.

We welcomed academic, research and industry contributions. The conference had the following tracks:

- Methodologies, techniques and algorithms
- Applications and services

We take here the opportunity to warmly thank all the members of the PESARO 2026 technical program committee, as well as all the reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and effort to contribute to PESARO 2026. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

We also thank the members of the PESARO 2026 organizing committee for their help in handling the logistics and for their work that made this professional meeting a success.

We hope that PESARO 2026 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in the areas related to performance, safety and robustness in complex systems. We also hope that Venice provided a pleasant environment during the conference and everyone saved some time to enjoy the historic charm of the city.

PESARO 2026 Chairs

PESARO Steering Committee
Wolfgang Leister, Norsk Regnesentral (Norwegian Computing Center), Norway
Mohammad Rajabali Nejad, University of Twente, the Netherlands
Rémy Houssin, Université de Strasbourg - ICube Laboratory, France
Yulei Wu, University of Exeter, UK

PESARO Publicity Chairs
José Miguel Jiménez, Universitat Politecnica de Valencia, Spain
Francisco Javier Díaz Blasco, Universitat Politècnica de València, Spain
Ali Ahmad, Universitat Politècnica de València, Spain
Laura Garcia, Universidad Politécnica de Cartagena, Spain
Sandra Viciano Tudela, Universitat Politecnica de Valencia, Spain

PESARO 2026

Committee

PESARO Steering Committee

Mohammad Rajabali Nejad, University of Twente, the Netherlands
Rémy Houssin, Université de Strasbourg - ICube Laboratory, France
Yulei Wu, University of Exeter, UK
Wolfgang Leister, Norsk Regnesentral, Norway

PESARO 2026 Publicity Chairs

José Miguel Jiménez, Universitat Politecnica de Valencia, Spain
Francisco Javier Díaz Blasco, Universitat Politècnica de València, Spain
Ali Ahmad, Universitat Politècnica de València, Spain
Laura Garcia, Universidad Politécnica de Cartagena, Spain
Sandra Viciano Tudela, Universitat Politecnica de Valencia, Spain

PESARO 2026 Technical Program Committee

Mohammad AlMasri, Nvidia, USA
Ehsan Atoofian, Lakehead University, Canada
Kaustav Basu, Arizona State University, USA
Morteza Biglari-Abhari, University of Auckland, New Zealand
Chérifa Boucetta, University of Reims Champagne Ardenne, France
Lelio Campanile, University of Campania Luigi Vanvitelli, Italy
Pasquale Cantiello, University of Campania Luigi Vanvitelli, Italy
Sowmya Chintakindi, ConglomerateIT, USA
Frank Coolen, Durham University, UK
Faten Fakhfakh, National School of Engineering of Sfax, Tunisia
Victor Flores, Universidad Católica del Norte, Chile
Rita Girao-Silva, University of Coimbra & INESC-Coimbra, Portugal
Marco Gribaudo, Politecnico di Milano, Italy
Christoph-Alexander Holst, inIT - Institute Industrial IT, Germany
Rémy Houssin, Université de Strasbourg - ICube Laboratory, France
Benoit Iung, Lorraine University, France
Christos Kalloniatis, University of the Aegean, Greece
Atsushi Kanai, Hosei University, Japan
Liuwang Kang, University of Virginia, USA
Sokratis K. Katsikas, Norwegian University of Science and Technology, Norway
Michel A. Kinsy, STAM Center | Arizona State University, USA
Vincent Latzko, Technische Universität Dresden, Germany
Wolfgang Leister, Norsk Regnesentral, Norway
Michele Mastroianni, University of Campania -Luigi Vanvitelli, Italy
Ilaria Matteucci, IIT-CNR, Italy

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

How to Assess the Technology Readiness Levels of an AI-based System? 1
Stephanie Gourdin, Pierrick Richard, and Juliette Mattioli

How AI Impacts the Digital Products and Services Performance in a Critical Context? 7
Benoit Huyot, Juliette Mattioli, and Xavier Bec

An End-to-End Trustworthy Knowledge Graph Engineering Methodology 13
Emna Amdouni, Lucas Mattioli, Faouzi Adjed, Afef Awadid, Martin Gonzalez, Loic Cantat, and Juliette Mattioli

Evaluation of Robustness, Reliability, and Safety of an Artificial Intelligence Based System 19
Lucas Mattioli, Annia Abtout, Martin Gonzalez, Afef Awadid, Kevin Mantissa, Faouzi Adjed, Joseph Machrouh, Jaime De Oliveira, Christophe Guettier, Hatem Hajri, and Juliette Mattioli

Defining a Minimal Set of Trustworthy Properties for Reliable Knowledge-Based Systems 25
Florence de Grancey, Gaelle Lortal, Claire Laudy, Amandine Audouy, Florent Chenevier, and Joshua Salort

What IF: Ultimate Intelligence FORGIVES? 33
Sharron Frammingham

Harnessing Trustworthiness in LLM Agents through Embedding Trustworthy Engineering Life-Cycles into System Prompts 39
Sabrina Chaouche, Lucas Mattioli, Frederic Barozzi, Raphael Braud, Faouzi Adjed, and Martin Gonzalez

Evaluating Performance, Safety, and Robustness of an AI-Based Airport Delay Alerting Tool with Calibrated Machine Learning for Operational Decision Support 47
Soufiane Momtaz, Otmane Idrissi, and Joseph Machrouh

Behavior Driven Performance Testing via Integrated MLOPs Pipeline 55
Bharath Kumar Maganti

Assessing Prediction Reliability for Probabilistic Pose Estimation 60
Omar del-Tejo-Catala, Javier Perez Soler, Nicolas Garcia Sastre, Pau Garrigues Carbo, Jose Luis Guardiola, Alberto Perez, and Juan-Carlos Perez-Cortes

Building Confidence: An Ontological Approach to Assurance of Safety-Critical Systems 66
Odd Ivar Haugen

Artificial Intelligence Contributions to Extending the Current Limitations of Virtual Reality for Integrating Operator Safety in Early-Stage Industrial Machinery Design 76
Remy Houssin and Amadou Coulibaly

How to Assess the Technology Readiness Levels of an AI-based System?

Stéphanie Gourdin
Thales Defense Mission Systems,
Elancourt, France
stephanie.gourdin@fr.thalesgroup.com

Pierrick Richard
Thales Defense Mission Systems,
Elancourt, France
pierrick.richard@fr.thalesgroup.com

Juliette Mattioli
Thales SA, cortAIx
Palaiseau, France
juliette.mattioli@thalesgroup.com

***Abstract*—The Technology Readiness Level (TRL) framework provides a rigorous yet adaptable structure for evaluating the maturity of Artificial Intelligence (AI) based systems, ensuring that advancements progress from theoretical research to operational deployment in a measured and transparent manner. Unlike traditional technologies, AI systems demand a more nuanced assessment due to their reliance on dataset and knowledge-base quality, AI model adaptability, and dynamic performance under varying conditions. In the early research phase (TRL 1–3), the focus lies in defining foundational elements such as operational boundaries, data governance, and preliminary compliance with regulatory standards like the AI Act, thereby mitigating future risks of non-compliance or technical shortcomings. As development advances into TRL 4–6, validation extends beyond laboratory settings to real-world environments, where AI components must demonstrate not only functional correctness but also robustness, scalability, and hardware compatibility. The final deployment phase (TRL 7–9) emphasizes full-system integration, ethical alignment, and sustained operational reliability, ensuring that AI solutions meet both technical and regulatory benchmarks before widespread adoption. In this paper we discuss the adjustment needed in the TRL evaluation for AI based systems and share a structured checklist-based tool to support this process by defining criteria and advisory risk-mitigation measures, fostering a balanced approach to innovation while preventing costly oversights.**

Keywords*- *AI system; Technology Readiness Level; TRL criteria; AI maturity assessment

I. Introduction

High-quality software products and computer systems are crucial to stakeholders. Quality models, quality requirements, quality measurement, and quality evaluation are standardized within the International Standards on SQuaRE [1]. The Technology Readiness Level (TRL) framework [2][3] offers a systematic approach for assessing the maturity of technological innovations, providing a common language that bridges the gap between technical development and strategic decision-making. By establishing clear, measurable criteria for assessing progress, the TRL scale enables organizations to communicate effectively across disciplines, ensuring that engineers, project managers, and executives share a unified understanding of where a technology stands in its development lifecycle. This shared framework not only facilitates more informed discussions about technical maturity, but also enhances the ability to conduct comprehensive risk assessments, allowing stakeholders to identify potential challenges and dependencies before they become critical issues [4]. Furthermore, the TRL approach serves as a valuable guide for investment decisions, helping organizations allocate resources more efficiently by distinguishing between technologies that require additional development and those that are already mature enough for deployment. In this way, the framework prevents the premature introduction of unproven solutions while simultaneously avoiding excessive investment in technologies that have already reached their full potential.

In order to guarantee the various trust properties of an AI-based system throughout its lifecycle [5][6][7], it is important to assess its TRL. Unlike conventional deterministic software, AI systems exhibit emergent behaviors derived from data and knowledge rather than explicit programming. Without rigorous validation, their performance characteristics and potential failure modes cannot be fully understood. Given the rapid evolution and unique challenges of AI development, it is crucial to adapt and refine maturity assessment methods such as the Technology Readiness Level (TRL) framework to ensure the responsible and effective deployment of AI technologies.

Today, some standards define maturity criteria [8][9], but to our knowledge there is no practical guideline on how to carry out such an assessment in practice. In this article, we present a dedicated tool developed to facilitate the adaptation and application of the TRL framework for Artificial Intelligence (AI) systems. The first section provides a brief overview of the TRL scale and discusses the reasons why adjustments are necessary when applying it to AI systems. Next, we describe the methodology used to derive the proposed framework and demonstrate its use in practice through our tool. We then detail the AI-specific adjustments made at each main stage of the TRL framework—namely, the research phase, development phase, and deployment phase—before concluding.

II. Technology Readiness Level framework in the context of Artificial Intelligence

The nine-level TRL scale provides a common language for evaluating and communicating the development status of technologies, from initial concept through full operational deployment [10]. The scale progresses through three broad phases:

- Research Phase (TRL 1-3): Basic principles are observed and reported (TRL 1), technology concepts are formulated (TRL 2), and proof of concept is established through analytical or experimental means (TRL 3). This phase focuses on fundamental research and feasibility studies.
- Development Phase (TRL 4-6): Component validation occurs in laboratory environments (TRL 4), followed by validation in relevant environments (TRL 5), and

culminating in system demonstration in relevant environments (TRL 6). This phase involves prototyping, testing, and integration of components into increasingly realistic conditions.

- Deployment Phase (TRL 7-9): System prototypes are demonstrated in operational environments (TRL 7), actual systems are completed and qualified through testing and demonstration (TRL 8), and finally, systems are proven through successful mission operations (TRL 9). This final phase represents the transition from development to full operational capability.

The framework provides a clear roadmap for technology progression, facilitating risk management, resource allocation, and alignment with technical and strategic objectives. This is particularly relevant in sectors where safety and reliability are critical, such as aerospace, defense, cybersecurity, and digital technology. For Artificial Intelligence (AI) systems [11], a TRL assessment is not just a formality, but a key factor in determining feasibility, risk and investment decisions.

The EU AI Act defines "*AI systems as machine-based systems operating with varying autonomy, adapting over time, and generating influential outputs, from predictions to decisions*". The field of AI mainly follows three major paradigms. The first is *data-driven AI*, which covers statistical and connexionist AI such as Machine Learning (ML) and Generative AI. Inspired by biological neural networks, this sub-discipline has dominated since the early 2020s due to its ability to deduce patterns from data. Secondly, *knowledge-based AI* (also known as symbolic AI) relies on knowledge representations such as ontologies and conceptual graphs. The distinction between the first two paradigms lies in how knowledge is acquired: data-driven AI extracts it automatically from examples, whereas symbolic AI encodes it explicitly through human expertise. Finally, *hybrid AI* encompasses any synergistic combination of various AI techniques, which could be enhanced by prior knowledge (such as mathematics, physics, or geometry).

Unlike traditional engineering systems, where TRLs are often tied to physical prototypes and testing, AI systems introduce unique complexities due to their adaptive, data-driven and frequently opaque nature. For example, AI-based systems may demonstrate exceptional performance within the controlled parameters of a laboratory setting, achieving metrics that suggest a high level of readiness [12][13]. However, when exposed to the unpredictability of real-world conditions, such as adverse weather, adversarial threats or the variability of human behavior, their effectiveness can diminish considerably. AI-specific failure modes must be carefully addressed before AI algorithms are deployed. For instance, data-driven AI models can become miscalibrated due to subtle shifts in data distribution during deployment, causing them to overestimate their predictive accuracy. The disparity between controlled testing and real-world application underscores the pivotal role of performance and TRL assessments. These provide a more accurate understanding of a technology's readiness.

The disparity between AI performance in controlled lab settings and real-world applications highlights the need to manage stakeholder expectations effectively. These expectations are frequently influenced by overly optimistic forecasts or marketing claims rather than concrete evidence. TRL assessments [14] provide an unbiased evaluation of an AI system's capabilities, ensuring that stakeholder expectations align with actual performance. This alignment fosters greater transparency and builds trust in the technology's potential.

TRL evaluations also guide Research and Development (R&D) by identifying the specific challenges that hinder an AI system's transition from theoretical success to practical implementation. For instance, a medical diagnostic AI tool may demonstrate high accuracy when analyzing historical patient data, suggesting advanced readiness in a controlled environment. However, when deployed in real clinical settings, it may encounter unexpected obstacles, such as compatibility issues with hospital infrastructure or usability limitations, that reduce its operational effectiveness. By pinpointing these gaps, TRL assessments enable organizations to prioritize efforts toward overcoming critical barriers, whether through improving system interoperability, enhancing user experience, or refining the technology's scope to more feasible applications.

Furthermore, the regulatory and ethical implications of AI deployment underscore the importance of rigorous TRL evaluations, particularly in high-stakes sectors where safety and security are paramount. Applications like autonomous weapons or AI-driven medical diagnostics demand thorough validation to ensure compliance with ethical and regulatory standards. For example, an AI-based credit scoring system must not only prove technical proficiency but also meet strict fairness, robustness, and transparency requirements before being responsibly integrated into financial institutions. The TRL framework offers a structured method for assessing these multifaceted dimensions, ensuring AI systems achieve the necessary certifications and public trust. This approach helps organizations navigate complex regulatory landscapes while mitigating risks associated with premature or inadequately validated deployments.

III. Methodology for developing TRL evaluation criteria for AI

The goal of this work is to establish clear criteria to objectively determine each TRL level, as the interpretation of the TRL scale can vary. While specific criteria could also be useful for evaluating classical algorithms, experts have generally relied on their experience and a shared understanding to assess maturity without a detailed scale. In contrast, the use of new AI-based algorithms—whether data-driven or knowledge-based— necessitates a reevaluation of how algorithm maturity is assessed and calls for more precise and tailored criteria.

Indeed, since these systems are less reproducible and data-dependent, there is a greater reliance on testing methodology and environment: applying classical criteria alone may lead to an overestimation of the maturity level. For example, using real data for both training and testing in volumes and with representativeness that are insufficient to ensure proper

algorithm generalization can give misleading results. Some AI-based algorithms can produce convincing results very quickly, but their maturity also depends on the maturity of the testing environment. Therefore, we propose to precisely detail the TRL criteria to ensure the actual maturity of algorithms. Here, maturity is defined as the ability to use the algorithm within a system without risk of unexpected or untested behavior.

The methodology to develop the AI TRL scorecard was based on various sources proposing approaches for measuring TRLs in classical algorithms, used by different institutions to define the meaning of the different levels on the scale (including documents from French Department of Defense, NATO, the European Union, and the ISO 16290 standard [8]). From this work, it emerged a list of criteria that can be used to measure TRL levels for classical algorithms more precisely than by merely interpreting the level definitions alone.

We based our work on Thales' dual experience: on one hand, in our algorithmic studies aimed at integrating Artificial Intelligence into our products within the demanding defense context; on the other hand, through our participation in the Confiance.ai program [15], which highlights critical issues that may arise during the maturation of algorithms whose definitions are not derived from explainable problem modeling but rather from learning. This work has allowed us to adapt and supplement the list of criteria to be considered specifically for the challenges posed by Artificial Intelligence.

The tool helps project teams assess algorithm maturity and identify early risks. Using a checklist in an Excel template, it shows the current TRL level and highlights areas needing improvement. It also serves as a record to document progress and support maturity claims.

Mandatory criteria directly contribute to TRL definitions by characterizing algorithm maturity; failure to meet any mandatory criterion indicates the associated TRL level is not achieved. These criteria align with ISO 16290 requirements for TRL validation.

Conversely, recommended criteria function as risk mitigation checkpoints to ensure critical considerations are addressed timely, preventing potential deadlocks during further maturation phases that could necessitate solution redesign. While strongly advised to be reviewed at specified milestones, non-compliance with these does not invalidate the reported TRL. For instance it is recommended to pay attention to software licensing: during the research phase, the license used does not impact the progression in maturity, but it may prevent industrialization if incompatible. Similarly, regulatory compliance (e.g., AI Act) does not impede research but prohibits product deployment if breached. Therefore, it is advisable to address compliance issues as early as possible; otherwise, investments in the technology may turn out to be futile, as the solution could turn onto a non-deployable technology.

IV. Methodology for Assessing AI TRL

To assess the Technology Readiness Level (TRL), it is essential to clearly define and baseline the technology or subsystem being evaluated. This involves specifying what is being assessed, its intended use, and its boundaries—a definition that becomes more detailed as the system matures. For higher TRLs, it is also necessary to clearly state the performance requirements and understand the mission, the system context, and the operational environment involved [8].

The evaluation tool includes a dedicated section for describing the element under assessment, with the level of detail adapted to the maturity phase (research, development, or deployment). For Artificial Intelligence projects, this means specifying both the algorithm's objectives and the details of its training process, especially the datasets used. Since AI development is highly iterative, with frequent changes even late in the process, TRL assessment for AI must also be dynamic and ongoing. Regular updates to the training database are common to improve the system's robustness and alignment with real-world conditions.

A TRL score alone does not provide an exact measure of the remaining effort or costs, especially for AI, where initial development (such as dataset creation and labeling) can be resource-intensive, but subsequent updates may be efficiently revalidated through automated testing frameworks.

Achieving TRL 9 requires reproducible algorithm behavior, which is often challenging for AI systems that naturally incorporate randomness or adaptivity. Small updates usually only impact specific criteria, and a detailed evaluation helps identify what must be retested versus what remains unaffected by changes. Adopting an iterative and flexible approach is key: accepting temporary regressions in TRL fosters continuous improvement and shorter validation cycles. Instead of focusing solely on linear TRL progression, this mindset supports faster refinements and helps accelerate industrial maturity, balancing readiness assessment with ongoing enhancement.

V. Research Phase (TRL 1-3)

In the classical interpretation of TRLs, the early research phase (TRL 1–3) focuses on identifying fundamental principles, defining an application concept, and demonstrating technical feasibility through a proof of concept. For conventional software or hardware systems, readiness at these levels is primarily assessed through the identification of system inputs and outputs, the definition of functional requirements, and the verification that the proposed solution complies with applicable standards and regulations. However, when dealing with AI-based systems, these criteria must be significantly extended to account for the data-driven and probabilistic nature of learning-based components or the stochastic nature of certain knowledge-based AI or hybrid AI approaches.

At TRL 1, beyond simply identifying system inputs and outputs, it becomes necessary to characterize the nature of the data and knowledge itself [16]. We recommend assessing early on, starting at this initial stage, whether the underlying AI technics principles comply with AI-specific regulatory constraints, such as those outlined in the AI Act [17]. This early evaluation helps avoid investing effort in developing systems that may later prove non-compliant.

% Complete	44 %		TRL 3	Analytical and/or experimental demonstration of the feasibility of critical functions
—	100	☑	Mandatory	At least one feasible application is identified for the specified technological concept and associated basic principles.
—	100	☑	Mandatory	The functional chain of the technology is established
—	0	☐	Mandatory	If the algorithm is data-driven, the data lifecycle is established: creation (data + labels), storage, data quality, and accessibility. If it is knowledge-driven, the knowledge lifecycle is established
—	100	☑	Mandatory	The evaluation database covers the domain of application of the algorithm in its core functions.
—	0	☐	Mandatory	Simulations and/or laboratory experiments on subsets have justified the feasibility of critical elements of the technology for the intended application.
—	100	☑	Mandatory	Performance measurement metrics are established, covering both AI-specific metrics and operational metrics of the target application. Metrics ensuring test statistics are included.
—	0	☐	Mandatory	These simulations have confirmed/clarified/optimized the expected performance magnitudes.
—	0	☐	Mandatory	The lifecycle of AI algorithm design is mastered, and the ability to reproduce performance is achieved.
—	0	☐	Mandatory	The algorithm's complexity is calculated with a view to implementation goals compatible with target constraints.
		☐	Recommended	Performances gains over previous and competing technologies are confirmed.
		☐	Recommended	Major risks and blocking points are identified (technology integration, environments, etc.).
		☐	Recommended	A document justifying choices or a study report is documented. (A document justifying choices or a study report is documented.)
		☐	Recommended	A potential user (client, product line) has expressed a preliminary interest in one of the identified applications within the scope and level of generality decided. This user will subsequently be referred to as the "client."
		☐	Recommended	The metrics for algorithm explainability are identified.
		☐	Recommended	The metrics for robustness and/or the availability of a monitoring tool for the algorithm are identified.
		☐	Recommended	The skills and processes (methods, tools, manufacturing, etc.) necessary for the development of the technology are identified.
		☐	Recommended	A roadmap towards concrete applications is outlined.

Figure 1. Criteria for TRL3

TRL 2, traditionally focused on defining the application domain and conceptual system architecture, must explicitly introduce the Operational Design Domain (ODD) [18] and formalize the processes for obtaining, qualifying, and governing data and knowledge, since their availability and representativeness directly impact system feasibility. This stage is achieved through initial simulations aimed at demonstrating the concept, without yet addressing representativity.

At TRL 3, where a proof of concept is established, readiness for AI-based systems cannot be inferred solely from functional demonstrations. Instead, it requires explicit evidence that evaluation criteria tailored to AI-based behavior have been formally defined, including performance metrics that capture AI system effectiveness, uncertainty, and failure modes, and whose computation is reproducible across datasets and experimental runs [12]. These metrics must be supported by a structured data lifecycle encompassing data ingestion, curation, and labeling processes, and complemented by initial considerations regarding explainability and robustness, which are necessary to interpret and contextualize measured performance.

At the end of this research phase, the explicit label of the technology, the level of generality and the research and industry players developing the technology is fixed and are part of the context for the evaluation. Furthermore, the functional chain of the technology and the performance measurement metrics are also established and baselined.

Consequently, for AI-based systems, TRL 1–3 no longer represent a simple progression from concept to feasibility, but rather a structured reduction of uncertainty across data, learning behavior, and evaluation criteria.

We show in Figure 1 how the template is constructed for the example of the TRL3 which is a major step especially for AI systems.The example particularly illustrates a case where some of the criteria required to validate the TRL are met, but not all. In this case, the template allows measuring the level of TRL achievement as a percentage. The criteria shown in black are common to both classical and AI algorithms, while the criteria specific to Artificial Intelligence work are highlighted in orange.

VI. Development Phase (TRL 4-6)

In the classical TRL framework, the development phase spanning TRL 4 to TRL 6 corresponds to the progressive validation of system components and their integration under increasingly realistic conditions. TRL 4 focuses on the validation of individual components or subsystems within a laboratory environment, TRL 5 extends this validation to relevant environments that approximate operational conditions, and TRL 6 culminates in the demonstration of a system or subsystem prototype operating in a representative environment. For conventional systems, readiness at these levels is assessed primarily through functional correctness, interface compatibility, and the ability to integrate components into a coherent system architecture.

For AI-based systems, however, this development phase introduces additional and non-trivial maturity criteria that go beyond classical component validation. At TRL 4, validation is not limited to functional behavior in a laboratory setting: the AI architecture must be explicitly designed to be compatible with the target hardware constraints, such as computational

% Complete	0 %	TRL 6	Demonstration using a demonstrator or prototype in a highly representative environment
—	0 ☐	Mandatory	The operational environment of the system is known.
—	0 ☐	Mandatory	A set of technology requirements is developed with the client.
—	0 ☐	Mandatory	A prototype that meets all these requirements is created.
—	0 ☐	Mandatory	A satisfactory demonstration of the prototype in the main non-laboratory environmental conditions required by the client has been performed.
—	0 ☐	Mandatory	The client and the supplier agree on the representativeness of the client's demonstration needs.
—	0 ☐	Mandatory	The demonstration has validated the architecture, functions, integration of risk aspects, and system specifications and its subsystems.
—	0 ☐	Mandatory	An analysis of real-time constraints is conducted.
	☐	Recommendec	Specific requirements regarding robustness against cyber attacks (e.g., adversarial) are identified and included in the tests.
	☐	Recommendec	Procurement and manufacturing processes have been tested and validated through the prototype's experience feedback.
	☐	Recommendec	All technology elements, including its interfaces, are available or easy to develop without difficulty.

Figure 2. Criteria for TRL6

resources, memory, and latency requirements. Moreover, the AI architecture and its interfaces are formally specified, enabling controlled integration with non-AI components. A critical addition at this level is the creation and use of a dedicated validation or qualification dataset, distinct from training datasets or the knowledge base, to assess AI model behavior under controlled and repeatable conditions.

At TRL 5, the transition to a relevant environment requires a stronger stabilization of the AI component itself. In contrast to earlier stages where iterative design of the AI algorithm design remains central, the model is considered ready for deployment in the sense that its parameters are frozen, thereby fixing the AI capacity behavior. The evaluation dataset is explicitly determined based on client or stakeholder needs, reflecting operational expectations and acceptance criteria rather than research-driven objectives. Performance validation at this stage relies on simulation environments or real-world data that were not used during training, in order to assess generalization capabilities and to reduce the risk of overly optimistic performance estimates.

The TRL 6, detailed in Figure 2, further extends classical system demonstration by introducing security considerations that are specific to AI-based systems. In addition to demonstrating functional performance in a representative environment, readiness at this level requires that the system's robustness against adversarial threats is explicitly taken into account. This includes assessing the susceptibility of the AI model to adversarial inputs or malicious perturbations and evaluating the impact of such attacks on system behavior [19]. Consequently, for AI-based systems, TRL 4–6 represent not only a progression toward system integration, but also a shift from AI-centric development to controlled deployment readiness, where architectural compatibility, dataset governance, model stability, and security become central indicators of technological maturity.

VII. Deployment Phase (TRL 7-9)

In the deployment phase, the objective is to transition from a prototype to a product that can be industrialized. The algorithms have already demonstrated their performance and robustness during the previous maturity ramp-up phases, and therefore no further modifications are made to them starting from this deployment stage. At this point, the Artificial Intelligence algorithms are fixed, and the process of moving from prototype to industrialization are identical to those applied to traditional software projects and do not require AI-specific adjustments.

VIII. Conclusion and Future Works

This paper presents a TRL assessment guide which offers a structured framework in order to guarantee the various trust properties of an AI-based system throughout its lifecycle. As operational contexts evolve, data-driven AI systems can degrade in ways traditional technologies do not. TRL assessment therefore verifies that systems are validated under sufficiently representative conditions, preventing premature advancement based only on selected benchmarks.

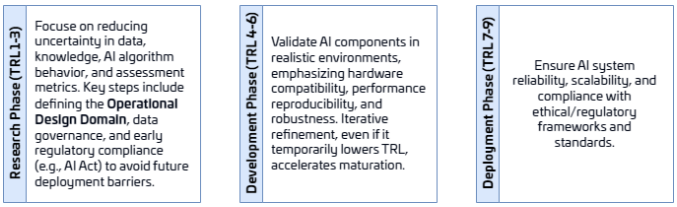


Figure 3. The TRL framework supports the AI system maturity assessment, bridging technical development and strategic decision-making.

As resumed Figure 3, lower TRLs address core capabilities, intermediate TRLs show subsystem integration, and higher TRLs validate fully integrated systems. AI components must also fit into complex existing architectures. The stepwise TRL progression ensures integration issues are found at the right development phase, not at deployment.

The TRL framework provides a common language for AI developers, users, acquisition authorities, and decision-makers. It helps prevent misalignment, such as laboratory demonstrations creating unrealistic expectations of operational

maturity, and is essential for governance and multinational collaboration.

This TRL framework is already used at Thales in various operational and research contexts [20]. It supports defining functional requirements and evaluation metrics during proposals, assessing the maturity of research components before transfer to Thales solutions, and ensuring that product and system maturity suits their intended defense and aeronautics uses. As noted, it is integrated into our engineering workflows to systematically reinforce AI governance. The tool is non specific to Thales and easy to use, therefore in the mid term, it will be made Open Source as part of the European Trustworthy AI Association[1], which provides open-source tools to support scalable and secure AI development.

The TRL scale nonetheless has limitations. Although it details the path from research to large-scale deployment, it does not inherently cover other factors critical to success. Regulatory compliance and market maturity add complexity: technologies must both function correctly and meet legal standards while being accepted in target markets. These issues are acute when moving between sectors, where differing interpretations and adaptations of TRLs can cause inconsistencies and misunderstandings.

In safety-critical fields such as aerospace and defense, where AI failures can be catastrophic, adherence to the TRL process is both best practice and an ethical imperative [21]. Each TRL stage functions as a checkpoint, requiring evidence of increased technological maturity before progression. This staged, evidence-based approach ensures rigorous validation under realistic conditions before moving from laboratory prototypes to operational use. It is especially crucial for systems that must satisfy certification standards like EUROCAE/SAE ARP6983 for aeronautics [22]. Emerging regulations, such as the European AI Act, also mandate phased development with progressive demonstrations of maturity, for which TRL assessments provide a natural structure, particularly in high-risk applications.

References

[1] ISO/IEC 25000:2014, *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Guide to SQuaRE*, Standard, 2014.

[2] J. Mankins et al., *Technology readiness levels*, 1995.

[3] S. Hirshorn et al., *Nasa systems engineering handbook*, 2017.

[4] S. Li et al., "A multi-dimensional assessment system for technology readiness levels", in *2017 4th International Conference on Systems and Informatics (ICSAI)*, IEEE, 2017, pp. 798–802.

[5] A. Awadid et al., "AI systems trustworthiness assessment: State of the art", in *12th International Conference on Model-Based Software and Systems Engineering*, 2024.

[6] K. Quintero et al., "An end-to-end method for operationalizing trustwothiness in AI-based critical systems", in *15th International Conference on Performance, Safety and Robustness in Complex Systems and Applications PESARO 2025*, 2025.

[7] E. Amdouni et al., "An end-to-end trustworthy knowledge graph engineering methodology", in *16th International Conference on Performance, Safety and Robustness in Complex Systems and Applications PESARO 2026*, 2026.

[8] *ISO 16290 : Space systems - Definition of the Technology Readiness Levels (TRLs) and their criteria of assessment*, Standard, 2014.

[9] B. Tucker and M. Bailey, "Rethinking technological readiness in the era of ai uncertainty", *arXiv e-prints*, arXiv–2506, 2025.

[10] J. Mankins, "Technology readiness assessments: A retrospective", *Acta Astronautica*, vol. 65, no. 9-10, pp. 1216–1223, 2009.

[11] J. Mattioli and C. Meyer, "Artificial Intelligence for Critical Systems", Thales, Tech. Rep., 2024.

[12] J. Mattioli et al., "An overview of key trustworthiness attributes and KPIs for trusted ML-based systems engineering", *AI and Ethics*, vol. 4, no. 1, pp. 15–25, 2024.

[13] J. Mattioli et al., "A brief overview of key quality metrics for knowledge graph solution illustration on digital notams", in *Proceedings of the AAAI Symposium Series*, vol. 7, 2025, pp. 206–213.

[14] A. Lavin et al., "Technology readiness levels for machine learning systems", *Nature Communications*, vol. 13, no. 1, p. 6039, 2022.

[15] Confiance.ai et al., *Towards the engineering of trustworthy AI applications for critical systems - the confiance.ai program*, 2022.

[16] J. Mattioli et al., "Information Quality: the cornerstone for AI-based Industry 4.0", *Procedia Computer Science*, vol. 201, 2022.

[17] L. Floridi, "The European legislation on AI: a brief analysis of its philosophical approach", *Philosophy & Technology*, vol. 34, no. 2, pp. 215–222, 2021.

[18] P. Koopman and F. Fratrik, "How many operational design domains, objects, and events?", in *Safeai@ AAAI*, 2019.

[19] M. Gonzalez et al., "Introducing RUM: A Methodological Contribution for Engineering Trustworthy AI Components in Industrial Systems", in *Proceedings of the AAAI Symposium Series*, vol. 7, 2025, pp. 153–160.

[20] *Thales AI in Real Use*, 2025. [Online]. Available: https://www.thalesgroup.com/en/advanced-technologies/artificial-intelligence.

[21] M. Cummings, "Rethinking the maturity of artificial intelligence in safety-critical settings", *AI Magazine*, vol. 42, no. 1, pp. 6–15, 2021.

[22] *ARP6983 - Process Standard for Development and Certification/Approval of Aeronautical Safety-Related Products Implementing AI*. [Online]. Available: https://www.sae.org/standards/arp6983-process-standard-development-certification-approval-aeronautical-safety-related-products-implementing-ai.

[1] https://www.trustworthy-ai-association.eu/

How AI Impacts the Digital Products and Services Performance in a Critical Context?

Benoit Huyot
cortAIx Factory SAS, France
benoit.huyot@thalesgroup.com

Juliette Mattioli
Thales SA, cortAIx, France
juliette.mattioli@thalesgroup.com

Xavier Bec
Thales Global Services SAS, France
xavier.bec@thalesgroup.com

***Abstract*—The Deliver Digital Products and Services (DDPS) framework is conceived to streamline the development and operation of digital-first solutions by reducing organizational silos and enhancing collaboration between operations engineering and product management. It integrates agile, DevOps, and "Shift Left" principles and is structured around five continuous, iterative activities: Explore & Plan; Build & Deliver; Release & Deploy; Operate & Support; and Performance Assessment & Learning. In light of the large-scale deployment of artificial intelligence (AI) in digital products and services, as well as the emergence of new AI-specific regulatory frameworks, the DDPS process requires adaptation. In particular, the EU AI Act introduces the notion of "intended purpose" as a central criterion for AI governance, tightly coupling liability and compliance obligations to the operational context of use. The development of trustworthy AI thus necessitates a holistic, lifecycle-oriented approach that integrates legal, ethical, and technical considerations across design, deployment, and operational phases.**

Keywords- Digital Products; Deliver Digital Products and Services (DDPS); DDPS Process; Trustworthiness Assessment.

I. INTRODUCTION

The "**Deliver Digital Products and Services**" (DDPS) process focuses on creating, distributing, and managing digital solutions, often AI-based, delivered mainly via online or networked environments rather than physical formats. These offerings include software, mobile apps, cloud platforms, streaming services, and digital marketplaces, designed to meet specific user needs in a fully dematerialized way. Unlike physical products that require production, shipping, or installation, digital products and services are delivered instantly and accessed anywhere with an internet connection, providing high convenience and flexibility.

Digital products and services typically run on public, private, or hybrid clouds, enabling automated distribution without local installation or hardware changes. Providers handle updates and scalability, matching capacity to demand and maintaining performance without physical reconfiguration. Automated deployment enables continuous, largely transparent roll-out of new features, security patches, and performance improvements with minimal disruption. These solutions are mainly operated and supported remotely using advanced monitoring, maintenance, and customer support tools [1]. Real-time tracking of performance, security, and system health enables proactive issue resolution. Support is provided via online channels such as chat-bots, help centers, and ticketing systems, delivering fast, efficient, contactless assistance while reducing costs and enabling scalable, round-the-clock support across regions.

The DDPS process applies only to these digital-first solutions, not to traditional or hybrid offerings that still rely on physical components or manual interventions. The framework is built to maximize efficiency, scalability, and user-centricity in a rapidly evolving digital landscape.

Trustworthy AI relies on legal, ethical, and technical foundations, reflected in requirements such as robustness, effectiveness, reliability, usability, human agency, and oversight [2]. It spans the entire AI lifecycle, from design and engineering to deployment and operation [3], and covers both technical systems and the actors and processes around them. This holistic view treats trust not only as a property of a digital product or service, but also as the result of relationships among stakeholders such as AI engineers, scientists, domain experts, and leaders. It focuses on understanding each stakeholder's perspective and maintaining trust as objectives, environments, and conditions change. The aim is to provide a practical framework for stakeholders to systematically assess and ensure the trustworthiness of AI-based DDPS.

After introducing in section III-A the DDPS activities [4], we will examine the integration of AI into digital products and services in the core of section III. It aims to support a robust design and delivery process that reliably generates effective solutions by rigorously meeting user and business needs, while ensuring compliance with applicable standards and regulations underlined in section II. This work builds on the results of the French initiative "Confiance.ai" which introduced a structured framework to assess the reliability of AI-based systems presented in section IV, especially those using machine learning, while simultaneously reducing the costs associated with over exploitation, engineering, and operational inefficiencies.

II. AI REGULATORY AND NORMATIVE LANDSCAPE

We first outline the AI regulatory and normative landscape we must follow. The EU AI Act defines a framework for high-risk AI systems to ensure safety, reliability, and ethical soundness [2]. High-risk domains include critical infrastructure, transportation (including aeronautics), medical devices, and essential public services. "High-risk" status follows a systematic assessment of a system's potential to harm human welfare or undermine fundamental societal values. A key feature is the presumption of conformity through harmonized standards, fostering trust, transparency, and accountability among AI developers, providers, and users [5]. Figure 1 presents the broad spectrum of standards for AI data, performance, and governance that underpin trustworthy and responsible AI.

Specific safeguards, including cybersecurity and functional safety measures, are central to regulations for high-risk AI sys-

 ISBN: 978-1-68558-395-8

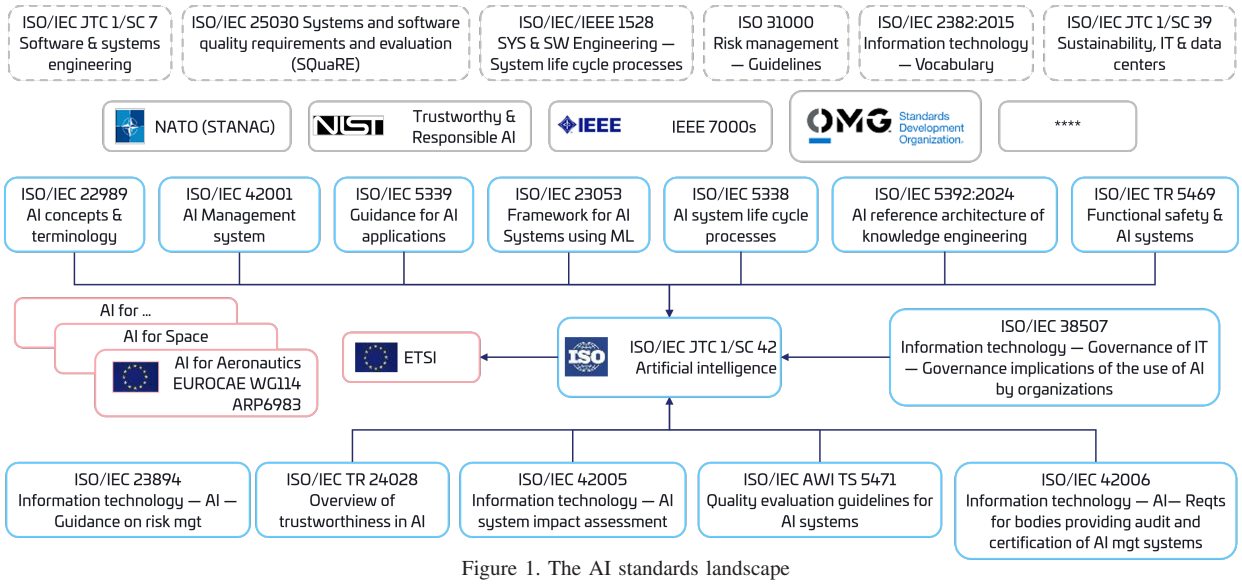


Figure 1. The AI standards landscape

tems. They go beyond conventional IT security by addressing AI-specific vulnerabilities through systematic risk assessments that identify and mitigate threats such as adversarial attacks, data poisoning or manipulation, and model drift. Unlike traditional digital services, AI-driven DDPS often exhibit emergent behaviors that cannot be fully predicted at the design stage. This unpredictability requires a fundamental rethink of the end-to-end design, development, and delivery of digital products and services to properly integrate AI components. The implications of this reconfiguration are examined in the next section.

III. Deliver Digital Products and Services Process

The engineering DDPS process (see figure 2) is conceptualized as a comprehensive end-to-end lifecycle framework that systematically enhances the cross-functional collaboration necessary to securely and efficiently design, develop, deploy, and operate digital products and services [6]. Its principal objective is to mitigate organizational and functional silos and to decrease both the frequency and complexity of handovers between operations engineering and product management.

A. Usual DDPS Workflow

The DDPS process embraces the concepts and culture of agility, DevOps and Shift left [4]. It is inspired by already existing concepts, Agile at scale frameworks, studies such as Team Topologies and Accelerate. It is founded on a "Continuous Everything" paradigm, within which five activities are executed on an ongoing, uninterrupted basis.

- "**Explore and Plan**": The objective is to continuously explore markets and customer needs, while also managing roadmap, capacity, budget, and organization.
- "**Build and Deliver**": Based on the roadmap, objective is to continuously take features and enablers from the Program Backlog and implement them in a continuous delivery process.
- "**Release and Deploy**": In alignment with the "Release on Demand" paradigm, the objective is to manage candidate releases throughout the entire delivery pipeline, culminating in their deployment into the production environment.
- "**Operate and Support**": The objective is to ensure continuous, uninterruptible service delivery and comprehensive technical support for end users.
- "**Measure and Learn**": Based on a data-driven culture, objective is to establish a framework for the implementation of continuous improvement, as well as ongoing learning and experimentation, both with regard to the quality of the products and services delivered and the optimization of the efficiency of operational activities.

Based on work on the life cycle of an AI based system [7] [8] and the workflow described above, we refined the overall DDPS process to integrate the induced issues of embedding AI [9].

B. Explore and Plan of an AI-based DDPS

The adoption of the EU Artificial Intelligence Act represents a significant milestone in the governance of AI, as it establishes the "**intended purpose**" as a foundational legal and conceptual criterion for delineating the scope, accountability structures, and regulatory compliance obligations applicable to AI systems. This notion encompasses the specific objectives of AI solution (AI-systems as well as AI-based digital products and services) deployment, the functional characteristics of these systems, and the technical and socio-organizational contexts in which they are conceived, designed, developed, and operated. By systematically linking liability regimes and compliance duties to the intended purpose, the AI Act constructs a comprehensive and robust regulatory framework that governs AI solution through a differentiated, risk-based approach. The primary objective of the "explore and plan" stage is to delineate the operational domain within which the AI-based digital product or service is required to function and to ensure that it

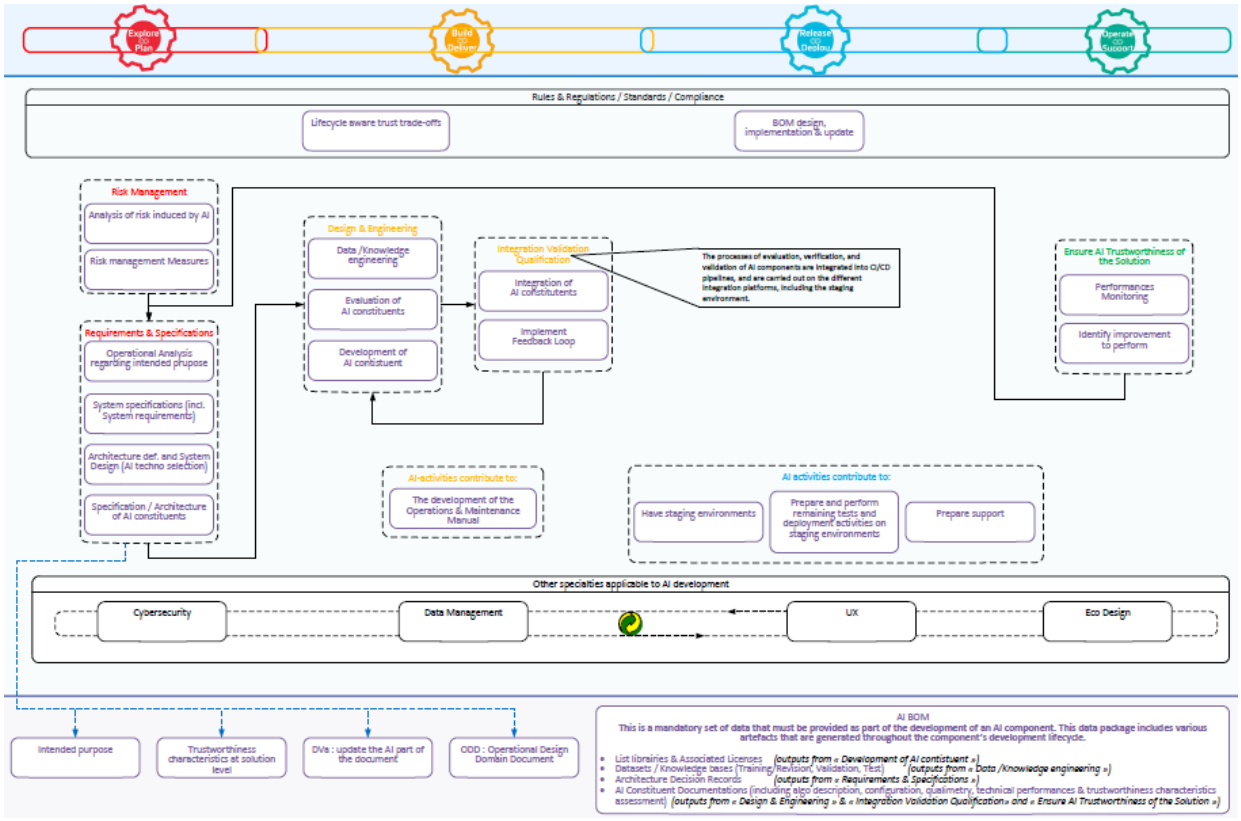


Figure 2. The AI-based Digital Products and Services Delivery Process

continues to operate as intended over time under all reasonably foreseeable operating conditions. This includes the systematic consideration of edge and corner cases, as well as potential failure scenarios, in order to mitigate unintended consequences that could endanger human life or compromise mission-critical functions.

Accordingly, the Intended Purpose Summary shall be articulated in a concise, clearly delineated statement specifying the system's functional capabilities, the entities or data upon which it operates, the intended user groups, and the operational context in which it is expected to be deployed. The Operational Design Document (ODD) [10] provides a formal description of the operational environment and the associated operating conditions. The development and analysis of the ODD have been initiated based on the preliminary formulation of the Intended Purpose. Thus, the "Risk Management" activity and the "Requirements and Specification" activity, which includes the ODD description, constitute the principal components of the "Explore and Plan" phase.

C. Build and Deliver of an AI-based DDPS

The aim of build and deliver is to produce AI constituents and integrate them into an operational version of the system. It includes design, documentation, implementation as well as integration and testing it results on a potentially releasable version of the system [11].

Once the risk analysis, the ODD, the functional architecture of the digital solution, and the associated technological choices have been finalized, the subsequent step consists in specifying each individual constituent. Multiple implementation streams, each dedicated to a specific constituent, may proceed in parallel, encompassing specification, development and implementation, unit-level evaluation, and delivery for integration. In addition, the design phase aims to examine the available data required to train the model and to define the functional and non-functional requirements of the AI constituent. These requirements should guide the design of the architecture of the AI-based digital product or service, the definition of the model-serving strategy, and the construction of a comprehensive test suite for the future AI constituent.

The subsequent phase, entitled "Integration, Validation, and Qualification," is dedicated to assessing the suitability of artificial intelligence (AI) techniques for the problem at hand through the implementation of a proof-of-concept for the AI component. In this phase, we iteratively execute several activities, including the identification and refinement of an appropriate AI algorithm for the target problem, as well as data engineering, knowledge engineering, and algorithm engineering. The principal objective of this phase is to produce a robust and reliable AI component that is ready for deployment [12].

D. Release and Deploy of an AI-based DDPS

The primary objective of this phase is to implement the solution incorporating one or more AI components by applying established DevOps practices, including systematic testing,

management and deployment of release candidates, continuous delivery, and operational monitoring.

Even when the majority of verification and validation activities are shifted left and executed on a continuous basis, certain activities must still be conducted immediately prior to release and deployment into the production environment. This phase includes the preparation and execution of the remaining tasks, such as operational readiness activities, deployment of the release to a staging environment, and subsequent validation and testing on that platform. Only upon successful completion of these activities is the release authorized for deployment into the production environment.

Certain AI constituent, particularly those grounded in data-driven approaches such as machine learning (ML), necessitate continuous operational monitoring to identify deviations during runtime. Consequently, the use of dedicated monitoring and management tools is mandatory [13]. The inherently static nature of trained ML models can lead to suboptimal performance in dynamically evolving environments. As a result, ML models must be capable of adapting to changes, including component wear and aging, as well as emerging data biases, in order to mitigate obsolescence caused by concept drift. Nonetheless, the specification of appropriate performance metrics for deployment monitoring is inherently problem-dependent. The recent proliferation and deployment of large language models (LLMs) further accentuate and amplify these challenges.

E. Operate and Support of an AI-based DDPS

The integration of AI components into products necessitates dedicated activities to systematically evaluate AI trustworthiness, with particular emphasis on the “Operate and Support” phase of the lifecycle. This phase encompasses all operational dimensions, focusing on day-to-day execution and customer service delivery, while ensuring that AI-enabled digital products and services consistently remain trustworthy.

To this end, a structured, formalized, and reproducible assessment framework is employed to evaluate key trustworthiness attributes, including effectiveness, reliability, security, validity, explainability, and accountability. The use of such a framework supports the continuous improvement of system quality and contributes to an enhanced overall customer experience [14].

As with cybersecurity, where an “Ensure Cybersecurity of the Solution” activity is embedded in operations, AI trustworthiness is maintained through a dedicated “Ensure AI Trustworthiness of the Solution” sub-activity [15]. During operations, this sub-activity continuously monitors and assesses AI performance, reliability, security, and validity so the solution keeps meeting its objectives. Monitoring mechanisms enable early detection and anticipation of issues such as model drift, helping to avoid or reduce service disruptions [13]. Explainability and traceability mechanisms further support efficient maintenance. When deviations, deficiencies, or non-compliance with trustworthiness requirements are found, corrective actions—such as model retraining, updates, or technical/procedural fixes—are initiated and tracked via the backlog to maintain alignment with defined trustworthiness criteria.

IV. Trustworthiness Performance Assessment

Assessing trustworthiness in AI-based products and services is a multifaceted challenge that goes beyond traditional metrics such as model accuracy or computational efficiency, requiring a holistic, systemic approach to ensure reliability, safety, and alignment with human and societal values [16]. Trustworthiness is a dynamic attribute that must be continuously monitored, validated, and adapted throughout the entire lifecycle of an AI-based digital product or service. This complexity stems from AI’s inherent uncertainty, stochastic behavior, and emergent risks, which differ from traditional software systems. Therefore, trustworthiness assessment must consider interactions between technical components, human oversight, and operational environments, all of which shape overall performance and dependability. This engineering activity, grounded in a data-driven culture, supports continuous improvement—a core value of the DDPS process—and is transversal to the four previous activities. Its main purpose is to continuously identify, track, and implement improvement actions by defining and reviewing KPIs relevant to the digital program. This step corresponds to the "Measure and Learn" activity in the DDPS process.

Assessing trustworthy performance requires shifting from a model-centric view to a Product- or Services-level one. Traditional AI evaluation focuses on model metrics like accuracy, precision, or recall on curated datasets, but this is inadequate for high-stakes applications. What matters is the real-world behavior of the digital product or service, not the model alone. This demands integration testing to track how uncertainties propagate through the solution, how human–machine interfaces support effective oversight, and how deterministic cybersecurity protocols can override AI outputs when they breach laws or ethical constraints.

Another key part of assessing trustworthiness is evaluating performance metrics, which must align with the real-time demands of high-risk applications [17]. Computational efficiency, particularly latency, where delays in processing could lead to failures. The worst-case execution time must be rigorously guaranteed to ensure that the system responds within safe operational windows.

A key challenge in assessing reliability is ensuring that training and validation datasets or knowledge bases are representative, since AI products are only as reliable as the data or knowledge used to design them. A model may perform well in the lab or on specific datasets but fail in real-world deployment, where variability, edge cases and environmental noise differ from the original data. This divergence occurs when the operational design domain (ODD)—the range of conditions in which the system is expected to operate—is not adequately reflected in the design data or knowledge. Reliability assessment must therefore rigorously validate information quality (data and knowledge) against real-world distributions [18] and test the digital solution’s ability to remain stable and predictable within its ODD.

TABLE I. MAIN KEY PROFILES

Role	Responsibilities	Key Skills
AI Engineers	Design, train, and optimize AI models. Implement robustness enhancements (*e.g.,* cyber attack or misuse).	MLOps/AIOps, Computer Science, Algorithm engineering
Data Analyst	Curate and preprocess datasets. Ensure data quality and representativeness.	Data Cleaning, Feature Engineering, Bias Detection.
AI Scientists	Develop and validate new AI constituents taking into account trustworthiness characteristics such as transparency and/or explainability.	Data-driven AI, Knowledge-based AI, Hybrid AI, Explainable AI (XAI), Visualization Tools
Domain Experts	Support Operational analysis regarding the intended purpose. Contribute to the Operational design domain definition	Domain Knowledge (*e.g.,* Defense, Aerospace, Cyber & Digital).
AI Leader	Ensure the AI system aligns with regulatory and ethical standards. Conduct bias audits and fairness assessments.	Recognized AI expert, AI discipline, master the regulatory framework, standards & Digital Ethics Charter
End Users	Provide real-world feedback on solution performance and usability through the feedback loop instanciated in the solution.	Domain-Specific Knowledge, User Experience (UX) Feedback.
AI Security Experts	Assess vulnerabilities specific for AI (*e.g.,* model inversion attacks, data poisoning). Implement defensive mechanisms (*e.g.,* differential privacy).	AI cyber security, Prompt Injection, Watermarking, etc.
Human Factors Engineers	Design user interfaces that present AI decisions clearly and intuitively. Ensure human oversight is effective.	UX/UI Design, Human-Computer Interaction (HCI), Cognitive Engineering.

AI products' stochastic variations in outputs, even when identical, complicate trustworthiness evaluations as they exhibit a non-deterministic nature. This makes regression, testing, certification, and auditing challenging due to the need for consistency to demonstrate compliance with safety standards. Ensuring repeatability is essential for safety-critical applications, where variations could lead to harmful consequences. Reproducibility is crucial for maintainability, certification, and auditability, mandatory in regulated industries like healthcare and aviation.

AI-based solutions are more complex because their dependability must hold not only technically but also under unforeseen failures and changing environments. Unlike traditional software, whose dependability can often be guaranteed, AI products and services rely on statistical performance trade-offs and require a risk-based approach to maintain safety and cybersecurity. Availability must be backed by fallback mechanisms that trigger when the AI encounters out-of-distribution inputs or low-confidence cases. Reliability covers not just uptime but stable performance over time, especially under concept drift. Safety and security require verifying that the system avoids catastrophic risks. Maintainability is critical due to AI's data-driven nature, demanding continuous retraining and adaptation to stay effective.

The usability of AI systems now includes multiple dimensions beyond traditional user-friendliness. Transparency, adaptability and alignment with human cognitive and ethical expectations are now key factors in trustworthiness assessments. Even if a system is technically proficient, it may still fail if its decision-making processes are opaque or does not offer explanations for its actions. Human-in-the-loop is a critical component of trustworthiness in areas such as healthcare diagnostics, autonomous vehicles and cybersecurity.

The governance and ethical dimensions of trustworthiness assessment emphasize the need for accountability, fairness and compliance. AI systems must be designed and deployed in a manner that respects fundamental rights, avoids discriminatory biases and ensures transparency. Standards such as ISO/IEC 42001 provide a structural framework for implementing governance mechanisms. Continuous monitoring facilitated by MLOps and AIOps frameworks is essential to detect performance degradation, data drift or ethical deviations over time; wher MLOps (resp. AIOps) focuses on the operationalization of ML (resp. AI) models, ensuring that they are deployed efficiently and maintained effectively in production environments. In contrast, ML/AI Engineering is primarily concerned with the development and the maintenance of an AI-based solution.

This lifecycle governance ensures that AI solutions remain aligned with their original design specifications and adapt to evolving realities without compromising robustness, explainability, reliability or fairness. Finally, the link between technical performance and human factors is key to trustworthiness assessment.

V. CONCLUSION

Integrating AI into digital products and services introduces technical and non-technical challenges. Major efforts aim to resolve these and enable early, cost-effective, and safe industrial AI deployment. The main challenges are: (i) trustworthiness—the product's ability to reliably deliver the expected service; and (ii) industrial efficiency—achieving this trustworthiness within acceptable cost and resource limits. Addressing these requires revising engineering practices to account for AI-specific characteristics and requirements. In this context, the paper re-examines the DDPS process with AI components integrated throughout its lifecycle.

But AI is not limited to the domain of computer science; it is also enabling the emergence and transformation of professional

roles across multiple sectors [9]. Some of the principal positions that are being newly created, especially in Thales's professional families or substantially redefined through the adoption of AI include data and AI scientists, AI engineers, AI security specialists, etc. (see Table. I).

For example, data and AI scientists are primarily concerned with the development and formalization of AI components, whereas AI engineers focus on the full spectrum of activities required to operationalize these components and deploy them in practical settings. It is often advantageous for these roles to be treated as distinct specializations, functioning collaboratively within a team, with clearly differentiated and complementary skill sets applied accordingly. Key profiles such as AI engineers, domain specialists, legal and compliance experts, and human–machine interaction designers—must jointly assess risks, transparency needs, and mechanisms for effective human oversight. Under the fully applicable EU AI Act [2], high-risk AI systems must offer full traceability to support accountability and regulatory supervision. This goes beyond traditional documentation, requiring continuous, tamper-proof audit trails across the AI lifecycle.

Our approach has been tested on a concrete use case: digital NOTAM [19]. A NOTAM (Notice to Airmen) is a standardized system providing pilots with time-critical information on airports, airspace, navigation aids, and other facilities affecting flight safety and operations. Conventional NOTAMs use a telegraphic, highly abbreviated format that is often ambiguous but still interpretable by humans. Digital NOTAMs, by contrast, must formalize aeronautical information for unambiguous machine processing while preserving all operationally significant nuances. Developing a trustworthy digital NOTAM therefore requires a rigorous process to ensure operational efficiency by reducing processing time, resource use, and reliance on manual interpretation; and safety and accuracy [20], by reducing ambiguity and misinterpretation and improving the reliability of flight operations data. Because Thales operates in aerospace, digital identity, defense, and security, this creates requirements for governance, operational controls, and technology infrastructure, and thus for new engineering artefacts such as the mandated AI Bill of Materials (AI-BOM), which provides traceability and detailed documentation of libraries, datasets, architectural decisions, and AI component specifications. This DDPS process is already used at Thales in various operational and research contexts and is integrated into engineering workflows to systematically reinforce AI governance.

REFERENCES

[1] K. Hui and P. Chau, "Classifying digital products", *Communications of the ACM*, vol. 45, no. 6, pp. 73–79, 2002.

[2] L. Floridi, "The European legislation on AI: a brief analysis of its philosophical approach", *Philosophy & Technology*, vol. 34, no. 2, pp. 215–222, 2021.

[3] L. Giraldo et al., "White Paper Trustworthiness For AI in Defence: Developing Responsible, Ethical, and Trustworthy AI Systems for European Defence", European Defence Agency (EDA), Tech. Rep., 2025.

[4] P. Müller, *Integrated engineering of products and services*. Fraunhofer Verlag, 2014.

[5] H. Sohier et al., "The Engineering of AI Evaluation and Scoring: Overview and Insights", in *2025 IEEE International Systems Conference (SysCon)*, IEEE, 2025, pp. 1–8.

[6] J. De Sordi et al., "Development of Digital Products and Services: Proposal of a Framework to Analyze Versioning Actions", *European Management Journal*, vol. 34, no. 5, pp. 564–578, 2016.

[7] K. Quintero et al., "An end-to-end method for operationalizing trustwothiness in AI-based critical systems", in *15th Int. Conf. on Performance, Safety and Robustness in Complex Systems and Applications*, 2025.

[8] L. Mattioli et al., "Evaluation of Robustness, Reliability, and Safety of an Artificial Intelligence Based System", in *16th Int. Conf. on Performance, Safety and Robustness in Complex Systems and Applications*, 2026.

[9] L. Korada, "AIOps and MLOps: Redefining Software Engineering Lifecycles and Professional Skills for the Modern Era", *Journal of Engineering and Applied Sciences Technology. SRC/JEAST-388. DOI: doi. org/10.47363/JEAST/2023 (5)*, vol. 271, pp. 2–7, 2023.

[10] T. Myklebust et al., "Definition of the system, operational design domain, and concept of operation", in *The AI Act and The Agile Safety Plan*, Springer, 2025, pp. 19–27.

[11] M. Adedjouma et al., "Engineering Dependable AI Systems", in *2022 17th Annual System of Systems Engineering Conference (SOSE)*, IEEE, 2022, pp. 458–463.

[12] J. Mattioli et al., "AI Engineering to Deploy Reliable AI in Industry", in *2023 Fifth International Conference on Transdisciplinary AI (TransAI)*, IEEE, 2023, pp. 228–231.

[13] F. Kaakai and P. Raffi, "Towards Multi-Timescale Online Monitoring of AI Models: Principles and Preliminary Results", in *SafeAI, AAAI's Workshop on Artificial Intelligence Safety*, vol. 3381, 2023.

[14] E. Popkova, "Quality of Digital Product: Theory and Practice", *International Journal for Quality Research*, vol. 14, no. 1, p. 201, 2020.

[15] A. Awadid et al., "Towards engineering processes to guide the development of trustworthy ml systems", in *2024 IEEE International Symposium on Systems Engineering (ISSE)*, IEEE, 2024, pp. 1–6.

[16] A. Awadid et al., "AI Systems Trustworthiness Assessment: State of the Art", in *Workshop on Model-based System Engineering and AI, 12th International Conference on Model-Based Software and Systems Engineering (Modelsward)*, 2024.

[17] J. Mattioli et al., "Towards a holistic Approach for AI Trustworthiness Assessment based upon Aids for Multi-Criteria Aggregation", in *SafeAI - The AAAI's Workshop on Artificial Intelligence Safety*, vol. 3381, 2023.

[18] J. Mattioli et al., "Information Quality: the Cornerstone for AI-based Industry 4.0", *Procedia Computer Science*, vol. 201, 2022.

[19] J. Mattioli et al., "A Brief Overview of Key Quality Metrics for Knowledge Graph Solution Illustration on Digital NOTAMs", in *Proceedings of the AAAI Symposium Series*, vol. 7, 2025, pp. 206–213.

[20] A. Awadid et al., "Reframing the System Engineering Lifecycle for AI Systems: An Intended Purpose-Driven Approach", in *9th International Conference on Software and System Engineering (ICoSSE 2026)]*, 2026.

An End-to-End Trustworthy Knowledge Graph Engineering Methodology

Emna Amdouni
IRT SystemX,
Palaiseau, France
emna.amdouni@irt-systemx.fr

Lucas Mattioli
IRT SystemX, Onera
Palaiseau, France
lucas.mattioli@irt-systemx.fr

Faouzi Adjed
IRT SystemX,
Palaiseau, France
faouzi.adjed@irt-systemx.fr

Afef Awadid
IRT SystemX,
Palaiseau, France
afef.awadid@irt-systemx.fr

Martin Gonzalez
IRT SystemX,
Palaiseau, France
martin.gonzalez@irt-systemx.fr

Loic Cantat
SafenAI,
Paris, France
loic@safenai.io

Juliette Mattioli
Thales SA, cortAIx,
Palaiseau, France
juliette.mattioli@thalesgroup.com

***Abstract*—Existing knowledge graph engineering methodologies provide limited support for governance, quality assessment, and accountability across the lifecycle, particularly in collaborative and industrial settings. This limits the use of knowledge graphs in safety-critical and high-risk AI systems subject to regulatory and ethical requirements, such as the EU AI Act. We propose an end-to-end Trustworthy Knowledge Graph (TKG) engineering methodology structured into three complementary dimensions: a methodology dimension covering KG construction phases, from knowledge elicitation and modeling to validation and deployment; a lifecycle dimension capturing continuous use and updates; and a transverse trustworthiness dimension integrating governance and quality assessment across all phases.**

Keywords-Trustworthy AI Engineering; Knowledge Graph's Lifecycle; Knowledge Graph Engineering; Trustworthiness Assessment.

I. Introduction

A. Trustworthy AI Engineering for Regulatory Compliance

As Artificial Intelligence (AI) grows in capability and scale, ensuring its reliability, robustness, and alignment with human intent is critical. Trustworthy AI Engineering [1] has emerged as a paradigm for developing and operating AI systems, especially in high-stakes and safety-critical domains [2]. This field integrates software engineering, systems engineering, cybersecurity, ethics, design, and cognitive science to address the complex challenges of modern AI. Its goal is to ensure AI systems are technically sound, accurate, ethical, transparent, compliant with regulations such as the EU AI Act, and resilient in uncertain, dynamic environments.

This emerging discipline tackles the uncertainty, limited reproducibility, and restricted verifiability of AI systems' behavior and decisions. Unlike conventional software, whose behavior can usually be deterministically specified and verified, many AI models function as "black boxes" [3], limiting assurances of consistency, safety, and value alignment. Deploying trustworthy AI in environments with ambiguity, non-stationarity, and adversarial threats demands a lifecycle approach spanning design, data engineering, deployment, monitoring, and maintenance, while embedding fairness, accountability, transparency, and robustness [4]. Trustworthy AI engineering therefore integrates advanced AI with safety, cybersecurity, reliability, ethical, and regulatory requirements to ensure systems are both innovative and compliant [5].

B. Key Concepts of Knowledge Graphs

Over the last decade, **data-driven AI** has become dominant, overshadowing symbolic AI. Connectionist and statistical methods mimic the brain's data-driven learning and excel at image and pattern recognition, but are limited in high-level reasoning, problem-solving and interpretability. **Knowledge-based AI** (symbolic AI) instead uses formal logic, rule-based systems and structured knowledge. In the Cartesian tradition, it defines intelligence via axioms, logical inference and domain expertise. Unlike connectionist systems, which learn implicit statistical correlations, symbolic AI encodes knowledge transparently, supporting precise reasoning, verifiability and adaptation to complex rule-governed settings.

In safety-critical domains such as aerospace, healthcare and industrial automation, marked by uncertainty and complexity, it is crucial to combine heterogeneous techniques. Integrating physics-based approaches (*e.g.*, differential equations and mechanistic models) with data-driven methods (*e.g.*, neural networks) yields **hybrid AI** that enables theoretically grounded, robust and adaptable decision-making.

A **Knowledge Graph** (KG) is a structured representation of entities and relationships, typically modeled as a graph. It enables semantic integration, reasoning, and explainability, making it suitable for industrial AI systems [6]. KGs are commonly formalized using the Resource Description Framework (RDF), where knowledge is expressed as triples (subject, predicate, object). For example, "*Symbolic AI is a subdiscipline of AI*" can be represented as $(\text{Symbolic AI}, \text{subdiscipline}, \text{AI})$. This formalism enables machine-processable and semantically rich knowledge [7].

KGs are large networks of entities and relations that explicitly model connections across domains, unlike traditional databases. For example, a KG might not only store that "*AI engineering is a new discipline*" but also link it to related methodologies and tools. Modern KGs use NLP, machine learning, and data mining to automatically extract, refine,

and update their contents, distinguishing them from static ontologies or taxonomies [6]. Systems like Google's Knowledge Graph and Microsoft's Satori use web-scale extraction to continually expand their repositories. KGs are application-agnostic, supporting use cases from semantic search (*e.g.*, enriching search results with context) to decision-support (*e.g.*, recommending treatments from patient data and medical literature). By integrating heterogeneous data into a unified, queryable layer, KGs have become central to AI-driven analytics and generative AI.

A KG is a dynamic, graph-structured knowledge base that formally represents entities, their attributes, and relationships. Combining semantic richness, scalability, and automated knowledge acquisition, KGs turn raw data into actionable insights, driving advances in AI, data science, and beyond.

C. Outline of the Study

While the Confiance.ai program [8] offers a tool-based methodology for ML-based system engineering, this study targets trustworthy engineering for symbolic AI. We examine the lifecycle of KG-based systems, focusing on development phases and the integration of KG-specific qualification. We then review evaluation measures needed for KG qualification. This work complements the ML engineering body of knowledge of the "European Trustworthy AI Association" [9], which defines an end-to-end methodology for trustworthy ML-based AI engineering.

II. A Trustworthy KG Engineering Methodology

A. Limits of Existing KG Engineering Methodologies

The current state of the art in knowledge engineering lacks accurate methodologies to address knowledge explicability, traceability, auditability, versioning, and governance in KGs within a trustworthy AI industrial context.

Early R&D ontology-centric methodologies such as METHONTOLOGY (METH) [10] and NeOn [11] focused on knowledge elicitation, conceptual modeling, and ontology formalization. NeOn added flexibility by integrating alignment, modularity, refinement, and evaluation to support ontology and graph development for real use cases. These methods emphasize requirement analysis, conceptualization, formal representation, implementation, and evaluation, but offer limited support for operational lifecycle management, advanced semantic quality evaluation, knowledge graph (KG) maintenance, and governance.

The On-To-Knowledge Methodology (OKTM) [12] was among the first to adopt a lifecycle view of ontology engineering for industrial use, defining phases such as feasibility study, baseline ontology development, refinement, evaluation, and maintenance. However, it does not explicitly address traceability, trustworthiness, or governance. More recent industry-oriented approaches, such as LOT4KG [13], summarize the KG lifecycle into four stages: implementation, publication, maintenance, and update. This work focuses on constructing a KG from an ontology and on knowledge updating and change analysis, but does not adequately address validation, evaluation, KG usage and integration within AI systems, or governance. Trust-related aspects are also not explicitly defined as lifecycle objectives, despite their importance for industrial AI applications.

Across these R&D methodologies, core knowledge engineering activities (e.g., elicitation, modeling, implementation, publication, and update) are well addressed. However, three major gaps remain: absence of an explicit trust-by-design lifecycle, limited integration of governance and accountability roles, and lack of continuous monitoring mechanisms for trustworthiness during operational usage.

In industry, Neo4j, a leading graph database provider, offers technical pipelines for automated KG construction from heterogeneous sources such as text, tabular data, relational databases, and ontologies. A typical pipeline includes data ingestion, KG modeling, data mapping, KG construction, enrichment, maintenance, and usage [14]. However, these pipelines do not explicitly address KG validation and quality assessment and do not define a comprehensive end-to-end KG lifecycle, especially regarding monitoring and continuous governance.

We propose a KG-centric Trustworthy Knowledge Graph (TKG) framework that combines construction and lifecycle. Unlike ontology-driven approaches, it enables continuous KG evolution without requiring a predefined ontology, while integrating refinement, validation, usage, update, and governance to ensure trust. Ontology-level integration is left for future work.

B. Trustworthy KG Engineering Dimensions and Phases

We describe the TKG framework through three complementary dimensions: (i) a **methodology dimension** defining the phases required to build a KG version, (ii) a **lifecycle dimension** governing its continuous evolution, and (iii) a **transverse trustworthiness dimension** ensuring governance, traceability, and quality across all phases.

The methodology involves five **main actors (domain expert, KG developer, curator, publisher, and end-user)** who contribute to KG engineering.

We refer to *knowledge artifacts* as all outputs of KG engineering, including specifications, conceptual models, knowledge graphs, and underlying data sources.

The following phases belong to the methodology dimension and describe the construction of a single KG version.

- **Knowledge Elicitation**: *Lead*: Domain expert. *Participant*: KG developer. Identify the scope of the use case, operational requirements, user stories, trustworthiness attributes, and data sources in collaboration with domain experts.
- **KG Modeling**: *Lead*: KG developer. *Participants*: Domain expert(s). Design the semantic model based on elicited requirements, defining domain concepts, relationships, and constraints, as well as trust-related metadata. Such metadata captures the provenance of the information stored in the KG, including its origin, how it was produced, and who it was validated. It also supports

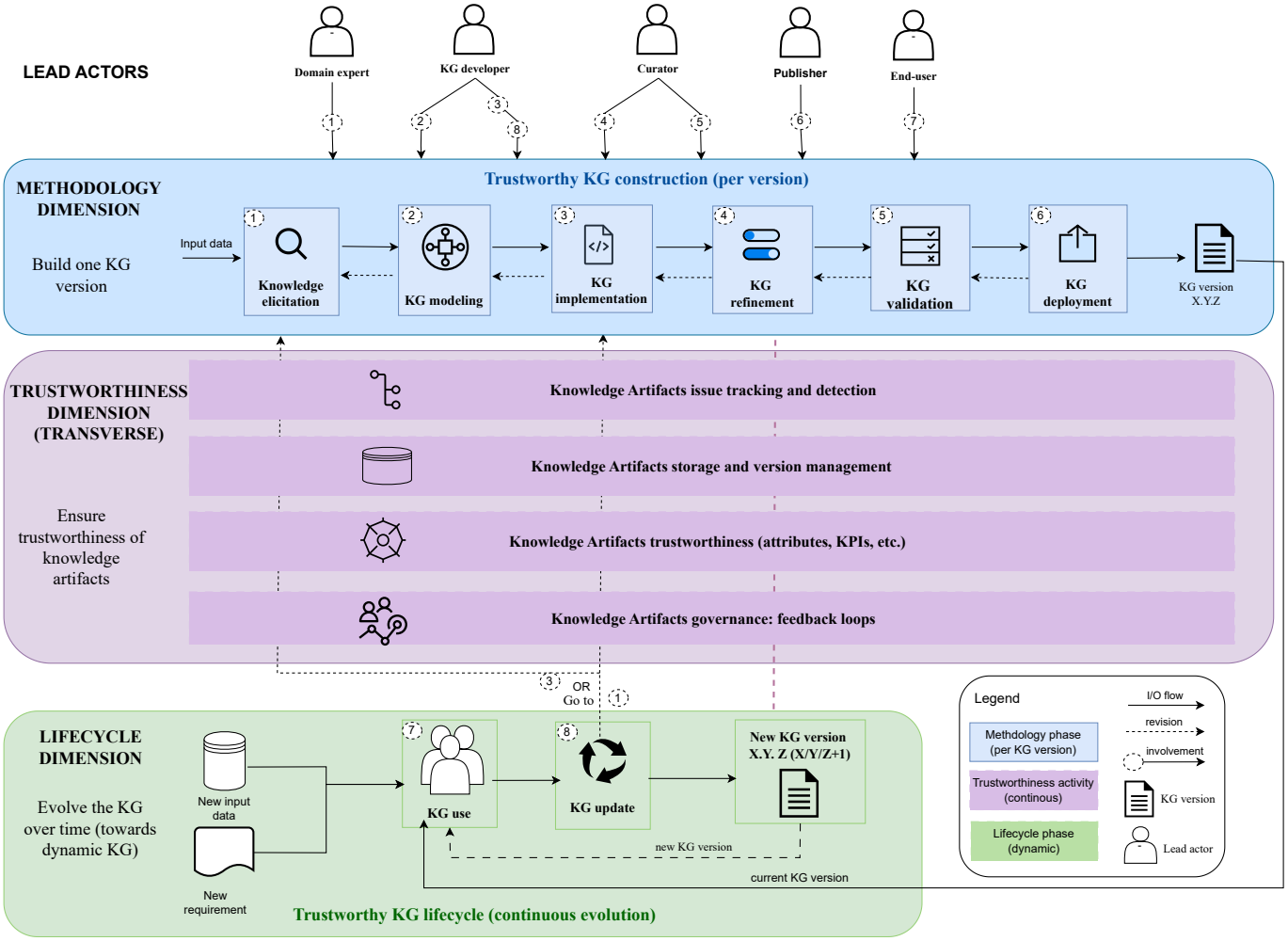


Figure 1. End-to-end TKG engineering with three dimensions: methodology (construction), lifecycle (evolution), and transverse trustworthiness (governance, traceability, and quality) applied to knowledge artifacts.

governance and maintenance processes. These aspects can be represented using established vocabularies such as Prov-O [15], Dublin Core Terms (DCT) [16], and the Data Quality Vocabulary (DQV) [17].

- **KG Implementation**: *Lead*: KG developer. Develop the RDF-based KG according to the conceptual model and agreed trust-related metadata (dct:source, prov:wasDerivedFrom, prov:wasGeneratedBy, etc.), ensuring traceability and version management.
- **KG Refinement**: *Lead*: Curator. *Participant*: KG developer and domain expert. Improve the quality of the resulting KG by adding new facts. This phase focuses on correcting errors, removing inconsistencies, improving the schema, merging duplicate entities and enriching relationships (better tying or clearer hierarchy). The refinement improves the correctness, consistency, precision and structure of what already exists in the resulted KG.
- **KG Validation**: *Lead*: Curator. *Participant*: KG developer and domain expert. Evaluate KG trustworthiness through KPIs and structured validation processes, including semantic, data, and expert validation. Verify compliance with constraints, detect inconsistencies, and document validation.
- **KG Deployment**: *Lead*: Publisher. *Participant*: KG developer. Deploy validated KG version through persistent URIs, machine-readable formats, and human-readable documentation to ensure transparency and reuse. In particular, define licensing, access policies, and usage governance to ensure transparency, reproducibility, and responsible reuse.
- **KG Usage**: *Lead*: End-user. *Participant:* publisher and curator. Knowledge graph usage should be transparent and controlled through secured APIs and catalogues. The objective of this phase is to control KG usage, track access logs, and detect new updates.
- **KG Update**: *Lead*: KG developer. *Participant*: curator. Incorporate new or changed data/requirements into the KG. It focuses on adding entities/relationships, removing out-dated facts, and adopting new changes. Updates are managed through trustworthiness mechanisms.

The trustworthiness dimension operates continuously across both the methodology and lifecycle dimensions, including

version management, issue tracking, KPI monitoring, provenance capture, and feedback loops. These mechanisms ensure auditability, accountability, and controlled evolution of the KG in industrial trustworthy AI systems.

C. Added Value of our Engineering Methodology

The proposed TKG lifecycle does not replace existing KG engineering methodologies, but extends them towards a trust-oriented engineering. As shown in Table I, key phases such as elicitation, modeling, implementation, and publication are already covered. We use the labels **"Explicit"**, **"Implicit"**, **"Limited"**, and **"Absent"** to indicate the level of support.

However, these phases are not integrated into a trust-oriented framework. Ontology approaches focus on defining knowledge schemas, while KG approaches focus on data construction and processing, without fully addressing trust across the KG lifecycle.

Our approach addresses this gap by introducing a KG-centric, lifecycle-oriented methodology structured into three complementary dimensions: a methodology dimension for building KG versions, a lifecycle dimension for their continuous evolution, and a transverse trustworthiness dimension.

III. Trustworthiness Attributes and Assessment

A. Trustworthiness Attributes

Trustworthiness attributes are fine-grained, measurable properties that define specific quality dimensions of an AI system. They clarify what constitutes trust in critical AI systems and fall into three capability areas: technical, usage, and governance. "Technical" covers verification of an AI component's validity and robustness; "Governance" concerns fundamental rights; and "Usage" addresses transparency, explainability, and usability. These attributes also cover relationships with third parties, especially quality assurance, audit, and certification.

Based on the EU AI Act and the European AI HLEG guidelines, trustworthy AI consists of six main requirements: robustness, effectiveness, dependability (including safety and security), usability, human agency (including transparency, interpretability and explainability), and human oversight (including ethical aspects). These characteristics are defined as follows:

- **Robustness** describes the system's ability to maintain its desired performance and functionality even when faced with challenging conditions, such as dealing with adversarial, uncertain, or imprecise inputs;
- **Effectiveness** is a measure of its ability to perform the functions necessary to achieve goals or objectives; it specifies the ability of a system to deliver a service that can be justifiably trusted;
- **Usability** denotes the degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a defined context of use.
- **Human agency** refers to the capacity of individuals to interact with, understand, and control the AI systems, ensuring these technologies are transparent, explainable, and aligned with human intentions;
- **Human oversight** encapsulates the evaluation and guidance of AI systems they operate within legal frameworks, fundamental rights, and general benevolence.

Trustworthiness can only be assessed when the Operational Design Domain (ODD) is clearly defined, specifying the conditions in which the AI system operates. Many AI prototypes fail to do this. Trustworthiness measures can identify issues before failures, support improvements in critical systems, and help designers build reliable, safe, and secure systems. No single assessment covers all trust dimensions, so trade-offs are required. Trustworthiness also spans the broader AI lifecycle, involving actors and processes such as engineers, operators, certification authorities, and insurance companies.

B. KG Effectiveness Assessment

In this section, we reuse the operational definition of correctness from [18] and summarize it for completeness. Among the six higher-level requirements introduced above, we focus on **effectiveness**, as it is most directly tied to KG content quality. Effectiveness measures how justifiably the KG can be trusted. To operationalize this notion, we decompose effectiveness into five complementary sub-dimensions, each with an associated metric.

Notation. Let:

- r denote the set of triples in the *produced* KG (assessed);
- r^* denote the set of triples in the *reference* KG (ground-truth);
- $r_{\text{crt}} = (r \cap r^*)|_{\Pi r}$ denote the *correct* subset of r, defined as the projection of the intersection $r \cap r^*$ into r, i.e., the triples in r that are confirmed as correct by r^*;
- $r_{cpt} = (r \cap r^*)|_{\Pi r^*}$ denote the *covered* subset of r^*, defined as the projection of the intersection $r \cap r^*$ onto r^*, i.e., the triples in r^* that are retrieved by r;
- $r \backslash r_{crt}$ denote the set of triples in r that are *not* confirmed by r^* (spurious or erroneous facts);
- $r^* \backslash r_{cpt}$ denote the set of triples in r^* that are *not* covered by r (missing facts).

All metrics take values in $[0, 1]$, where 1 indicates perfect performance on the corresponding dimension.

Correctness: Ensuring Factual Accuracy and Truthfulness - Correctness [19] concerns the factual accuracy and truthfulness of information encoded in the KG. It requires verifying that entities are correctly identified, relationships accurately reflect real-world connections, and attribute values match ground truth. It is the degree to which a produced answer matches the reference output *without introducing new spurious content*.Noted μ_{correct} [20] it is defined as $\mu_{\text{correct}} = 1 - \frac{|r \backslash r_{\text{crt}}|}{|r|} \in [0, 1]$, with $r_{\text{crt}} = (r \cap r^*)|_{\Pi r}$ being the KG resulting from the projection of the intersection of r and r^* on r and $r \setminus r_{\text{crt}}$ is the complementary of r in r^*

Completeness: Capturing the Domain Knowledge - Complementary to correctness, completeness [21] measures whether the system returns all required elements of the reference output. It reflects how fully the KG captures all relevant

TABLE I. COMPARISON OF ONTOLOGY ENGINEERING AND KNOWLEDGE GRAPH APPROACHES

Engineering Activities	Ontology Engineering			Knowledge Graph Approaches	
	METH	NEON	OTKM	LOT4KG	TKG
Elicitation / Specification	Explicit	Explicit	Limited	Explicit	Explicit
Conceptual modeling / Design	Explicit	Explicit	Implicit	Explicit	Explicit
Implementation	Explicit	Explicit	Explicit	Explicit	Explicit
Refinement / Enrichment	Absence	Limited	Explicit	Explicit	Explicit
Validation / Evaluation	**Implicit**	**Implicit**	**Limited**	**Limited**	**Explicit**
Deployment / Publication	Absence	Absence	Limited	Explicit	Explicit
Usage	**Limited**	Explicit	Explicit	**Absence**	**Explicit**
Update / Maintenance	**Limited**	**Limited**	**Limited**	Explicit	**Explicit**
Governance	**Absence**	**Limited**	**Limited**	**Limited**	**Explicit**

entities, relationships, and attributes within its intended scope. The completeness (denoted as μ_{complete}) measures the quantity of r contained in r^*. It is formerly defined as: $\mu_{\text{complete}} = 1 - \frac{|r^* \setminus r_{\text{cpt}}|}{|r^*|} \in [0, 1]$,

Consistency: Maintaining Internal Logical Coherence - Consistency assesses the internal logical coherence of the KG, ensuring that assertions do not conflict and that representations follow defined constraints and business rules [22]. It includes syntactic consistency, where structures follow established patterns, and semantic consistency, where relationship meanings remain uniform across the graph.

Logical Consistency quantifies the absence of contradictions: $\mu_{\text{LC}} = 1 - (\frac{|contradictions|}{|r|})$ measuring logical coherence by tracking contradictions discovered during reasoning processes relative to the total number of inferences made.

Representativeness: Faithfully Reflecting Domain Distributions - Representativeness concerns whether the KG accurately reflects the real distribution and characteristics of its domain. It focuses on coverage biases, ensuring minority cases are properly represented alongside dominant patterns and that the graph does not systematically favor certain entity or relationship types.

Timeliness: Maintaining Currency and Relevance - Beyond the core dimensions, Timeliness measures whether information remains current and relevant. Thus, a metric associated to timeliness $\mu_{\text{timeliness}}$ can be defined as [23] $\mu_{\text{timeliness}} = \max(0, 1 - \frac{currency}{volatility}) \in [0, 1]$, In the formula, $currency$ correspond to the age of the data when delivered to the user and $volatility$ is the length of time the data remains valid.

IV. ILLUSTRATION ON THE BODY OF KNOWLEDGE OF THE TRUSTWORTHY ML ENGINEERING

A. The Body of Knowledge

A body of knowledge (BoK) is "*structured knowledge employed by members of a discipline to inform their practice or work*" [24]. BoK design is widely used to define and model concepts through knowledge acquisition, fusion, storage, and retrieval. Knowledge is acquired from diverse data types by extracting entities, attributes, and relations, and is typically stored in KG databases. Thus, BoK design

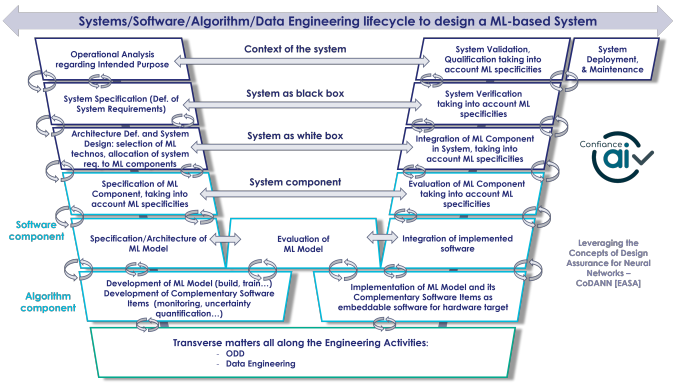


Figure 2. The view of the ML Engineering BoK - https://bok.confiance.ai/

entails developing a KG that provides users with comparable problem-solving capabilities. In our context, the BoK (see figure 2) serves as the "ground truth" for ML Engineering [25]. Its core components are the concepts, knowledge, skills, standards, terminology, guidelines, practices, and activities that define a field or specialization. It includes data repositories, performance indicators, and other tools to ensure reliable, trustworthy ML engineering, and spans multiple engineering domains [26].

B. Main Issues of Trustworthiness

As with any symbolic model, a BoK can only ever be an approximation of reality. New observations based on ML engineering use cases can inform the acquisition of further knowledge. Therefore, evaluating the accuracy of the knowledge represented with respect to reality is essential for creating an adequate model. These limitations are related to the symbol grounding problem [27], and concern the extent to which representational elements are hand-crafted rather than learned from data. Thus, several features must be taken into account when developing a BoK:

- **Redundancy**: Are there any knowledge items that are identical or equivalent to another (subsumed)?
- **Consistency**: Is there inconsistency, ambiguity or indeterminacy? Is it deliberate? Are there multiple outcomes?

- **Minimality**: Can the knowledge set be reduced/ simplified? Is the shortened version logically equivalent to the original?
- **Completeness**: Does the knowledge set include all entries?

A well-designed BoK should have: representational accuracy, to capture all necessary knowledge; inferential adequacy, to manipulate representations and generate new knowledge consistent with existing structures; inferential efficiency, to guide reasoning effectively by storing relevant information; and acquisition efficiency, to easily incorporate new knowledge, preferably through automatic methods.

Peer reviews were carried out with various stakeholders (data scientists, software and systems engineers, and cybersecurity and safety engineers, among others) to assess the appropriateness and quality of the acquired knowledge in relation to the end-to-end ML engineering methodology [28].

V. Conclusion and future works

This paper analyzes the structural and governance dimensions of knowledge graph engineering through a combined methodology and lifecycle perspective, enabling the continuous construction and evolution of KG-centric systems. A formal specification of input/output artifacts, activities, and tasks, as well as deeper investigation of dynamic aspects, remains future work. The CSIA program has adapted this lifecycle to AI engineering via the ML Engineering Body of Knowledge [25][26] and is systematically revising the BoK to rigorously assess and improve its consistency, minimality, and completeness. The proposed lifecycle extends existing KG engineering methodologies by embedding governance, quality assurance, and reliability requirements in all phases, as required for safety-critical AI systems. We also introduce a structured taxonomy of reliability attributes and explore quantitative techniques for evaluating KG effectiveness, enabling systematic, actionable, and measurable assessment.

Acknowledgment

This work has been supported by the French government under the "France 2030" program, as part of the SystemX Technological Research Institute within the CSIA Project.

References

[1] M. Adedjouma et al., *Towards the Engineering of Trustworthy AI Applications for Critical Systems. The Confiance.ai Program*, 2022.

[2] J. Perez-Cerrolaza et al., "Artificial intelligence for safety-critical systems in industrial and transportation domains: A survey", *ACM Computing Surveys*, vol. 56, no. 7, pp. 1–40, 2024.

[3] V. Hassija et al., "Interpreting black-box models: a review on explainable artificial intelligence", *Cognitive Computation*, vol. 16, no. 1, pp. 45–74, 2024.

[4] M. Poretschkin et al., "Guideline for Trustworthy Artificial Intelligence–AI Assessment Catalog", *arXiv preprint arXiv:2307.03681*, 2023.

[5] J. Mattioli et al., "AI Engineering to Deploy Reliable AI in Industry", in *5th Int. Conf. on Transdisciplinary AI (TransAI)*, 2023, pp. 228–231.

[6] D. Fensel et al., "Introduction: What Is a Knowledge Graph?", in *Knowledge graphs: Methodology, tools and selected use cases*, Springer, 2020, pp. 1–10.

[7] Z. Chen et al., "Knowledge Graph Completion: A Review", *IEEE Access*, vol. 8, pp. 192 435–192 456, 2020.

[8] K. Quintero et al., "An End-to-End Method for Operationalizing Trustwothiness in AI-Based Critical Systems", in *15th Int. Conf. on Performance, Safety and Robustness in Complex Systems and Applications (PESARO)*, 2025.

[9] European Trustworthy AI Association, *European trustworthy ai association*, Accessed: 2026-04-20, 2025. [Online]. Available: https://www.trustworthy-ai-association.eu/.

[10] M. López, "METHONTOLOGY: from ontological art towards ontological engineering", in *Proceedings of the…*, 1997.

[11] S. Singhania et al., "NeOn: News Entity-Interaction Extraction for Enhanced Question Answering", *arXiv preprint arXiv:2411.12449*, 2024.

[12] Y. Sure et al., "On-to-knowledge: Semantic web-enabled knowledge management", in *Web Intelligence*, Springer, 2003, pp. 277–300.

[13] R. Pernisch et al., "When ontologies met knowledge graphs: Tale of a methodology", in *European Semantic Web Conf.*, Springer, 2024, pp. 286–290.

[14] Neo4j, Inc., *Building knowledge graphs: A practical guide*, https://neo4j.com/developer/knowledge-graph/, Accessed: 2026-04-17, 2023.

[15] W3C Provenance Working Group, *Prov-o: The prov ontology*, https://www.w3.org/TR/prov-o/, W3C Recommendation, 2013.

[16] Dublin Core Metadata Initiative, *Dublin core metadata element set, version 1.1*, https://www.dublincore.org/specifications/dublin-core/dces/, DCMI Recommendation, 2012.

[17] W3C Data on the Web Best Practices Working Group, *Data quality vocabulary (dqv)*, https://www.w3.org/TR/vocab-dqv/, W3C Recommendation, 2016.

[18] J. Mattioli et al., "A Brief Overview of Key Quality Metrics for Knowledge Graph Solution Illustration on Digital NOTAMs", in *AAAI Symposium Series*, vol. 7, 2025, pp. 206–213.

[19] C. Laudy et al., "HLIF2024: a Competition for High-Level Information Fusion", in *2024 27th International Conf. on Information Fusion (FUSION)*, IEEE, 2024, pp. 1–8.

[20] C. Laudy and N. Museux, "How to evaluate high level fusion algorithms?", in *2019 22th International Conf. on Information Fusion (FUSION)*, IEEE, 2019, pp. 1–8.

[21] S. Issa et al., "Knowledge graph completeness: A systematic literature review", *IEEE Access*, vol. 9, 2021.

[22] J. Lehmann et al., "Quality assessment for linked data: A survey.", *Semantic Web (1570-0844)*, vol. 7, no. 1, 2016.

[23] O. Hartig and J. Zhao, "Using web data provenance for quality assessment.", *SWPM*, vol. 526, 2009.

[24] T. Ören, "Toward the body of knowledge of modeling and simulation", in *Interservice/industry training, simulation, and education Conf. (I/ITSEC)*, vol. 2005, 2005.

[25] J. Mattioli et al., "Leveraging Knowledge Graph to design the Machine-Learning Engineering Body-of-Knowledge", in *2024 Conf. on AI, Science, Engineering, and Technology (AIxSET)*, IEEE, 2024, pp. 258–265.

[26] J. Mattioli et al., "ML System Engineering Supported by a Body of Knowledge", in *Proceedings of the 16th International Joint Conf. on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, vol. 3, 2024, pp. 331–338.

[27] S. Harnad, "The symbol grounding problem", *Physica D: Nonlinear Phenomena*, vol. 42, no. 1-3, pp. 335–346, 1990.

[28] M. Adedjouma et al., "Engineering Dependable AI systems", in *2022 17th Annual System of Systems Engineering Conf. (SOSE)*, IEEE, 2022, pp. 458–463.

Evaluation of Robustness, Reliability, and Safety of an Artificial Intelligence Based System

Lucas Mattioli
IRT SystemX, Onera
France
lucas.mattioli@irt-systemx.fr

Annia Abtout
Talan,
France
annia.abtout@talan.com

Martin Gonzalez, Afef Awadid, Kevin Mantissa, Faouzi Adjed
IRT SystemX,
France
{*first-name.last-name*}*@irt-systemx.fr*

Joseph Machrouh, Jaime De Oliveira
Thales Land & Air Systems,
France
{*first-name.last-name*}*@thalesgroup.com*

Christophe Guettier, Hatem Hajri
Safran,
France
{*first-name.last-name*}*@safrangroup.com*

Juliette Mattioli
Thales SA, cortAIx,
France
juliette.mattioli@thalesgroup.com

***Abstract*—This paper proposes a paradigm-aware framework for evaluating AI robustness, reliability, and safety, arguing that current methods, designed for supervised learning, fail to reflect the diversity of modern AI. It distinguishes but links the three properties: robustness as performance under perturbations and distribution shifts; reliability as consistency and calibration over time; and safety as a broader socio-technical goal that depends on but goes beyond both. The main contribution is a systematic analysis of how these properties affect the AI system life cycle defined by ISO/IEC 5338, and how they vary across four paradigms: data-driven, symbolic, hybrid, and generative AI. Each paradigm has characteristic failure modes requiring tailored assessments—from adversarial testing and drift detection for neural networks, to formal verification for symbolic systems, to red-teaming and alignment for large language models. The framework embeds these assessments into systems engineering life cycles, stressing that reliability must be addressed at every stage, from requirements to post-deployment monitoring.**

Keywords- trustworthy AI; robustness; reliability; safety.

I. Introduction

Artificial Intelligence (AI) from major research labs and tech companies is now critical in high-stakes sectors such as aerospace, healthcare, automotive, and defense, where failures can be severe. This has spurred global efforts to define and ensure "trustworthy AI," as rigorously outlined by the European Commission's High-Level Expert Group [1]. Trustworthy AI covers human oversight, technical safety, privacy, transparency, fairness, societal and environmental responsibility, and accountability. In 2024, the EU AI Act made many of these requirements legally binding for high-risk AI systems, while organizations such as EASA and ISO/IEC created standards and certification frameworks for AI safety and reliability. The French program "Confiance.ai" [2] introduced a structured methodology for assessing machine learning trustworthiness, now maintained by the European Trustworthy AI Association, which offers open-source tools for scalable and secure AI development.

This paper introduces a paradigm-aware framework for evaluating the Robustness, Reliability, and Safety (RRS) of AI systems, contending that current, supervised-learning-centric methods overlook other paradigms (see Table I). It shows in section III how RRS concerns differ for data-driven, symbolic, hybrid, and generative AI, each requiring specific evaluations: adversarial testing and drift detection for data-driven models, formal verification for symbolic systems, compositional checks for hybrid models, and red-teaming and alignment assessments for generative AI such as large language models. Then in section IV, RRS is integrated across the AI lifecycle, from specification and design to validation, deployment, and monitoring, emphasizing continuous management in real-world use. The paper concludes in section VI that RRS are systemic properties of deployed AI, not merely features of trained models, and thus demand ongoing reassessment to preserve long-term trustworthiness.

II. Definitions and Conceptual approach

Building on the EU AI Act, the AI HLEG guidelines [1], the Confiance.ai methodology [3], and the EASA AI certification roadmap [4], we define AI trustworthiness through seven pillars: Robustness, Reliability, Safety, Explainability, Fairness, Privacy, and Governance (Table II). This article addresses only Robustness, Reliability, and Safety (RRS), which here have domain-specific meanings beyond conventional systems and software engineering. They are essential for trustworthy AI in critical domains such as healthcare, automotive, aerospace, defence, and security, and must be precisely specified to preserve intended behaviour across diverse scenarios.

A. Robustness, Reliability, and Safety Properties

In traditional engineering, **robustness** is a system's ability to perform reliably under minor disturbances. In AI, robustness must extend beyond adversarial attacks—crafted inputs that mislead models—to include distributional shift, where real-world data differs from training data and degrades performance. For example, a vision-based neural network trained in well-lit conditions may perform well in the lab but fail in dynamic environments. Such failures show that AI robustness requires

TABLE I. AI PARADIGMS W.R.T DOMINANT FAILURE MODES

Paradigm	Examples	Dominant Failure Modes
Data-Driven AI	Deep neural networks, SVM	Adversarial attacks, distributional shift
Symbolic AI	Expert systems, logic programming	Incomplete rule bases, logical inconsistencies
Hybrid AI	Neuro-symbolic, Physics/Geometry-Informed NN	Interface mismatches between constituents
Generative AI	LLMs, diffusion models	Hallucinations, prompt injection, bias

adaptation to varying contexts, not just resistance to noise, and that models appearing robust in tests may be fragile in practice, calling for evaluations beyond traditional ones.

We measure AI **reliability** using metrics such as failure rate, event rate, and error rate. An AI system is reliable if it consistently performs its intended function over time and under specified conditions. Assessment must include hardware failures, where physical components stop working (e.g., a faulty GPU), and software failures, where the system does not fulfil its purpose. Because failures may not halt use, we define 'failure' broadly, allowing multiple failures per unit. Reliability metrics then capture whether the system failed, time to failure, and, for recurring failures, the failure event rate.

Machine learning introduces further dimensions of reliability due to its probabilistic and opaque nature, including failure rates, output correctness, confidence calibration, and temporal stability [5]. A model may score well on a static test set yet still cause critical errors if its confidence is mis-calibrated. Reliability also degrades as data, concepts, or user behaviour change; for example, a predictive maintenance model may begin accurate but drift as its training data become unrepresentative. Reliability therefore demands continuous monitoring and recalibration throughout the system lifecycle.

Under ISO 26262, **safety** goes beyond harm prevention to cover AI-specific risks in dynamic, human-centred settings. In AI systems, safety is a property that reflects model performance, user interaction, misuse, and edge cases. It requires a holistic approach to human-AI collaboration, fail-safe mechanisms, and ethical alignment, supported by technical safeguards, organizational and procedural controls, and human-override protocols for transparent decisions and continuous, adaptive risk assessment and monitoring [6].

Robustness, reliability, and safety are interdependent. At the AI Constituent (AIC) level, robustness, uncertainty, and monitoring cannot generally be cleanly modularized, as they are all built, trained, and calibrated around the AIC's central model. The RUM Methodology [7] provides a principled basis for this view. Unlike model-centric evaluation, it treats AICs as atomic units whose behaviour must be assessed across their full lifecycle—specification, development, deployment, and updating—in line with end-to-end system engineering. It justifies treating AICs as indivisible and provides a structured set of mostly non-aggregative trust metrics to capture trustworthiness across the lifecycle. The framework also offers operational tools, such as AI Blueprints for runtime monitoring, human-in-the-loop interaction, and long-term maintainability, supporting trustworthy AI deployment in industrial settings.

B. Inter-dependencies of Robustness, Reliability and Safety

Robustness, reliability, and safety in AI are closely linked, enabling systems to work in both controlled and real-world settings. This RRS triad must be balanced and evaluated together: robustness handles variation and adverse conditions; reliability ensures stable, repeatable performance; safety limits behaviour to acceptable risk levels. Neglecting any one dimension can create vulnerabilities (*e.g.*, a model that seems reliable in testing may fail in deployment). As AI systems grow more complex and autonomous, jointly assessing these three aspects is vital for technically sound, trustworthy, and socially beneficial deployments.

Robustness is a prerequisite for reliability: A system must maintain performance under unexpected changes or disruptions. An AI model that withstands adversarial inputs, sensor noise, or data shifts is more reliable. Without robustness, performance can degrade unpredictably, causing inconsistent outputs and undermining even well-trained models when small input changes trigger cascading errors.

Reliability alone does not guarantee safety. Technical metrics matter little without systemic context. An AI can be highly reliable—consistently correct under defined conditions—yet still be dangerous if its objectives conflict with safety or operational standards. A language model, for example, may reliably generate fluent, relevant text while still producing harmful, biased, or misleading content if misaligned with safety constraints. Reliability is necessary but not sufficient for safety: a system can function flawlessly yet pose unacceptable risks if its goals are flawed or safeguards against misuse and unintended consequences are missing.

Safety is the culmination of these efforts, extending from technical guarantees to the socio-technical contexts in which AI operates. It is not inherent to a model but emerges from its interactions with environments, users, and systems. A system may perform well under normal conditions yet still cause disasters in rare edge cases—for instance, a healthcare diagnostic AI might excel in trials but become unsafe if clinicians over-rely on it without understanding its limits.

Safety demands a holistic approach that integrates technical robustness, human factors, organizational protocols, and ethics so AI aligns with human expectations, societal norms, and real-world complexity. Robustness, reliability, and safety are hierarchical yet interdependent: robustness resists disruptions, reliability sustains consistent performance, and safety embeds both in human-centered design. Neglecting any produces technically capable but untrustworthy systems, so AI must be evaluated on all three to remain safe and dependable in dynamic real-world settings [8].

TABLE II. ROBUSTNESS, RELIABILITY AND SAFETY ARE PILLARS OF AI TRUSTWORTHINESS

Pillar	Core Question	Key Evidence Types	Primary Standards
Robustness	Does the system maintain performance under perturbations, distributional shift, and adversarial inputs?	Adversarial test sets, OOD benchmarks, stress tests	ISO/IEC 24029, EASA DAL robustness requirements
Reliability	Does the system produce correct outputs consistently across its operational domain?	Performance metrics, failure rate analysis, uncertainty quantification	DO-178C, ISO 26262, IEC 61508
Safety	Does the system avoid causing harm to people, assets, or the environment?	Hazard analysis, FMEA/FMEDA, runtime monitoring logs	ARP 4754B, ARP 4761A, ARP 6983, MIL-STD-882

III. PARADIGM-SPECIFIC ASSESSMENT METHODS

Assessing an AI system's trustworthiness requires understanding its core properties, since AI paradigms differ in architecture, outputs, and failure modes. Four main paradigms have distinct features that shape their reliability.

A. Overview of AI Paradigms

Data-driven AI uses statistical methods like neural networks and evolutionary algorithms to learn from data. It performs well in controlled settings but struggles with real-world unpredictability due to sensitivity to data shifts and attacks. Even with explainability and robustness tools, a gap persists between controlled and real-world performance, leading to bias, failures, and vulnerabilities.

Symbolic AI uses human-readable knowledge bases and logical deduction for interpretability and formal correctness, but its reliability degrades with incomplete or inaccurate data. Missing rules, contradictions, and other gaps cause rigid reasoning. Traditional verification helps, yet maintaining complete, consistent knowledge bases is a core challenge.

Hybrid AI combines approaches such as neuro-symbolic models or constraint-informed neural networks to exploit their strengths. However, component interactions can introduce trust issues, as interface errors may compromise reasoning, even though symbolic constraints can also enhance trustworthiness by restricting outputs to logically valid conclusions.

Generative AI (GenAI), including large language models and diffusion-based image generators, uses vast datasets to create content, but its unpredictability makes it unreliable. It can produce factual errors, be manipulated, and its scale and emergent behaviour undermine traditional verification. Red-teaming and behavioural benchmarks offer partial safeguards, but full reliability remains out of reach.

B. Assessment Methods by Paradigm

RRS assessment in AI systems depends on the paradigm, as each has distinct behaviours, failure modes, and assurance needs. One-size-fits-all methods are ineffective; tailored approaches are essential. Without them, critical vulnerabilities may stay hidden, producing systems that work in controlled settings but fail in real-world use.

Data-driven AI learns statistical patterns from data rather than explicit rules, creating robustness, reliability, and safety challenges. Small input changes can trigger large output shifts (adversarial vulnerabilities), revealed through adversarial testing that simulates worst cases. Robustness certification checks that outputs remain stable within defined ranges. Distributional shift—when deployment data differs from training data—demands continuous monitoring to prevent degradation. Reliability is typically assessed with methods like cross-validation, while safety uses adapted frameworks such as System-Theoretic Process Analysis (STPA). However, most methods are statistical, not causal, limiting failure explanation and systemic risk prevention.

Symbolic AI uses explicit rules and logic for formal verification and deterministic reasoning, supporting robustness by avoiding adversarial vulnerabilities. It faces issues like logical inconsistencies and incomplete knowledge bases that undermine reliability. Formal methods (model checking, theorem proving) give strong correctness guarantees. Safety can benefit from traceable hazard paths, but depends on knowledge-base completeness. Symbolic AI offers strong formal guarantees, yet its effectiveness hinges on rule coverage and quality.

Hybrid AI combines data-driven and symbolic methods, exploiting their strengths while adding interface risks. Robustness requires validating how symbolic components interpret data-driven outputs. Reliability is evaluated via statistical testing plus formal verification, and safety via contract-based design. Embedding physics-informed or domain-specific constraints further improves robustness.

GenAI is challenging because of opaque, emergent behaviour and open-ended outputs. Traditional robustness tests are inadequate, as failures often reflect misalignment rather than adversarial noise. Prompt robustness testing examines how input variations affect outputs, exposing vulnerabilities like jailbreaking and prompt injection. Watermarking embeds detectable patterns in content to flag AI-generated misinformation [9]. Reliability covers accuracy, truthfulness, bias mitigation, and factuality, supported by metrics and bias audits. The main safety problem is alignment, keeping models beneficial and controllable. Techniques such as red-teaming and Reinforcement Learning from Human Feedback target value alignment but lack formal guarantees, leaving residual safety risks and highlighting the need for new assurance methods.

IV. INTEGRATION INTO SYSTEMS ENGINEERING LIFE-CYCLES

Trustworthiness cannot be added to an AI system after development; it must be built in from the outset as a system-level property spanning the entire lifecycle (see Figure 1 and Table III), from inception through operation and monitoring. This section explains how our approach aligns with established systems engineering practices, focusing on the EASA AI concept paper and ARP 6983 for aerospace [10], the ISO/IEC

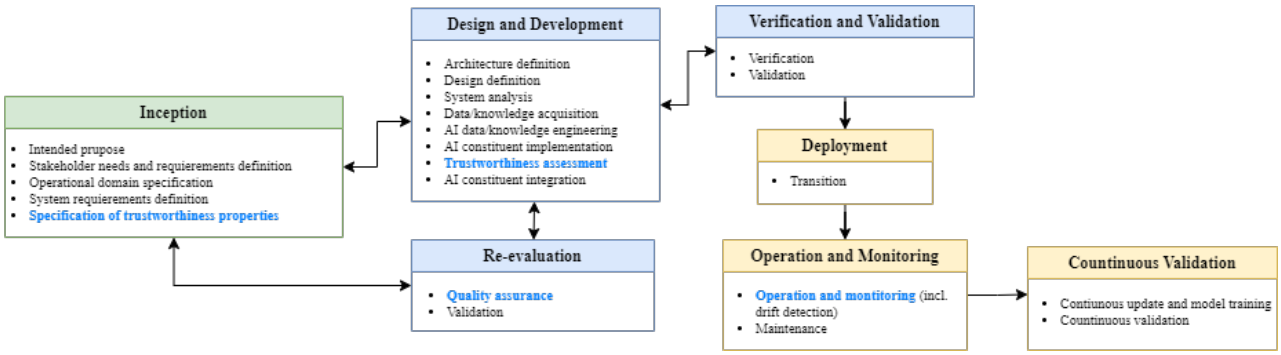


Figure 1. Variation of the ISO/IEC 5338 AI system life cycle processes with respect to robustness, reliability and safety requirements

TABLE III. THE RRS ACTIVITIES W.R.T. THE ENGINEERING PHASES

Phase	RRS Activities
Requirements	Define RRS goals (*e.g.*, "System must handle 95% of adversarial inputs without failure").
Design	Architect for assurance (*e.g.*, modularity for hybrid AI).
Verification	Apply paradigm-specific methods (*e.g.*, formal proof for symbolic AI).
Validation	Test in operational environments (*e.g.*, shadow mode for LLMs). Post-Deployment Continuous monitoring (*e.g.*, drift detection, red-teaming updates).

5338 standard for AI-specific software lifecycle processes[1], and the Confiance.ai methodology for industrial AI [11].

A. Inception Phase: Specifying Trustworthiness Properties

The Inception Phase captures high-level intents, refines them into detailed requirements, and decomposes tasks into executable units with planned components. It underpins all later assessment activities. The AI system must function reliably as intended and remain robust in unexpected situations. Robustness, reliability, and safety requirements must be expressed in measurable, testable terms; vague statements like "the system shall be robust" are not verifiable.

An AI system's **intended purpose** defines its function, outputs, users and performance limits. Set during Inception, it shapes verification and validation by specifying the system's role and acceptable behaviour. The **operational domain** (OD) defines the conditions under which an AI system is designed to function as intended [10]. The OD is part of the functional specification for trustworthiness because: 1) transparency about the OD clarifies system capabilities and limits (as required by the AI Act); 2) the OD is the reference domain for all operational trustworthiness attributes; and 3) the OD itself must be complete, consistent and human-readable. The OD is thus both a design constraint and an assessment tool: tests must verify performance throughout the OD, and inputs outside it must trigger defined safe behaviours.

- **Performance requirements**: quantitative reliability targets for each scenario class within the OD and operational needs, validated against the hazard analysis rather than generic engineering judgment.
- **Robustness requirements**: quantitative limits on acceptable performance degradation under defined perturbation types and magnitudes, including adversarial and distributional robustness.
- **Safety requirements**: constraints on AI outputs derived from system-level hazard analysis, expressed as invariants that must never be violated. For data-driven components these are probabilistic; for symbolic components they can be formal logical constraints verifiable by model checking.

The OD defines the system's environment, inputs, and operational constraints, reflecting the data used for training and validation. Intended purpose concerns outputs and overall function, while OD concerns inputs and conditions of use. Deviating from the intended purpose is misuse; operating outside the OD, even for an authorized purpose, creates out-of-distribution scenarios that can degrade performance without warning. The OD sets the system's operational boundaries; the intended purpose sets its accountability and functional limits.

B. Design Phase and Development: RRS Assessment

The design phase defines the system architecture to assess feasibility and later evaluation costs. Sound design principles can improve RRS and simplify its assessment.

- **Separation of concerns**: architectures that separate perception (data-driven), reasoning (symbolic), and actuation allow component-level assessment with paradigm-specific methods and clear interfaces, which is far more tractable than assessing a monolithic end-to-end neural system.
- **Explicit uncertainty representation**: propagating and exposing uncertainty estimates at all interfaces (rather than using point predictions) enables runtime monitoring and lets the symbolic reasoning component act conservatively when the data-driven component is uncertain.
- **Redundancy and diversity**: for high-assurance applications, architectural redundancy with diverse AI implementations can achieve required reliability and safety even when no single AI component can. Dissimilar redundancy—using different datasets, architectures, or paradigms—mitigates common-cause failures that defeat homogeneous redundancy.

[1]https://standards.globalspec.com/std/14651195/iso-iec-5338

- **Graceful degradation**: defining degraded operational modes (fall-forward to simpler, more reliable AI components; fall-back to human operators; fail-safe to a defined safe state) creates an architectural safety barrier that limits the consequences of AI component failure, regardless of failure mode.

C. Verification and Validation Phase

The AI validation and verification (V&V) process follows these principles:

- **Test plan coverage**: the test plan must explicitly cover the OD, sampling all identified scenario classes and oversampling safety-relevant minority classes. Any OD coverage gaps are safety-critical. Test set size must satisfy statistical power requirements for each metric, with documented sample size calculations.
- **Independence of V&V**: the V&V team must be independent from the development team to avoid optimistic bias, as required in safety-critical software standards. The V&V team must not have contributed to training data collection, model development, or hyperparameter selection.
- **Learning process verification**: for data-driven AI and GenAI, V&V must assess both the trained model and the training process. This includes verifying that the dataset meets documented quality requirements, the training pipeline is reproducible, hyperparameter selection did not contaminate the test set, and the trained model matches the documented architecture and configuration.
- **Formal analysis where feasible**: when architecture and computational budget allow, formal verification should supplement statistical testing. For symbolic components, it provides completeness guarantees; for hybrid components, compositional verification combines formal results for symbolic parts with statistical results for neural parts to obtain system-level guarantees.

D. Re-evaluation phase and post deployment monitoring

Pre-deployment V&V offers only point-in-time quality assurance, confirming the system meets requirements at deployment. For AI systems,this is insufficient as performance can degrade with changing operational environments. Continuous assurance, ongoing collection and evaluation of RRS evidence across the lifecycle, is therefore essential.

The **post-deployment monitoring** architecture must be defined in the system design. Key components include: drift monitors comparing operational to training data; monitors of accuracy and failure rates against operational ground truth; detectors for unusual outputs; and pipelines logging safety-relevant events. Monitoring is challenged by AI opacity and emergent capabilities, so improved anomaly detection and model evaluation are needed.

Performance degradation thresholds (quantitative performance drops that trigger alerts, degraded operation, or system withdrawal) must be specified in operational requirements and tied to the safety case. Any AI changes—retraining, dataset updates, configuration changes—must undergo change impact assessment to determine whether partial or full re-verification is required.

V. Examples

The **aeronautics industry**'s growing reliance on AI-based systems—from in-service support and autonomous flight operations to pilot decision support—demands unprecedented rigor in assessing robustness, reliability, and safety. Unlike traditional, deterministic or rule-based aviation software, data-driven or hybrid AI models learn from data, introducing variability that requires specialized validation. For example, machine learning algorithms for computer vision in aircraft inspection, such as Airbus tools that detect micro-cracks in composite materials, must be robust to adversarial inputs (manipulated images or sensor noise) that could cause false negatives and missed structural failures. Predictive analytics tools, like those Safran uses to anticipate engine component wear, must remain accurate with incomplete or biased historical data to support reliable maintenance decisions. Safety-critical avionics, including single-pilot operations and AI-driven flight optimizers (such as the TopSKy Sequencer by Thales [12]), need fail-safe mechanisms for edge cases like severe weather or conflicting air traffic control instructions to preserve passenger safety. Thus, even though traditional aviation standards (*e.g.* DO-178C and ARP 6983) provide partial guidance and EASA is developing a unified AI certification framework for aeronautics [10], dedicated RRS assessments are essential. Deep learning in pilot assistance tools, such as Airbus's Vision Based Landing Approach Runway Detection (LARD) [3], require continuous monitoring to detect 'black box' decision drifts. Without robust verification and validation—such as formal methods or stress testing on synthetic data—even well-trained models may fail in rare but plausible situations, including cyber-attacks.

In the **healthcare domain**, the growing use of AI systems, from data-driven diagnostic tools to generative models for clinical decision support, demands rigorous assessment of RRS. Unlike traditional clinical software based on deterministic rules, data-driven imaging models learn statistical patterns from training data, introducing variability that requires specialized validation. Deep learning algorithms for radiology or pathology must generalize beyond their development site; in practice, sensitivity to input noise, scanner differences, and population shifts often reveals brittleness in models whose internal validation had seemed adequate. Reliability is further weakened by confidence miscalibration, temporal drift in populations and imaging protocols, and inconsistent performance across operating conditions, all of which can cause systematic yet hidden errors. Safety, in turn, goes beyond technical performance: a diagnostic AI may excel in trials yet be unsafe if clinicians over-rely on it, misunderstand its limits, or lack human-override procedures for out-of-scope inputs.

For GenAI in clinical decision support, the RRS profile differs markedly from that of data-driven systems. Robustness is undermined by prompt sensitivity, as fabricated details in clinical prompts can cause models to elaborate on embedded errors; prompt-based safeguards mitigate but do not remove this

risk. Reliability must cover not only accuracy but also factual consistency and temporal stability, since reasoning failures are a primary source of errors and cast doubt on LLM trustworthiness across repeated high-stakes interactions. Safety arises from how model outputs integrate into clinical workflows: clinicians may over-trust fluent but wrong answers, and the lack of clear failure signals creates regulatory obstacles for approval as medical devices. This underscores the need for clinically tailored red-teaming, alignment protocols, and human-override mechanisms as structural assurance requirements.

Assessing robustness, reliability and safety is not only a technical task but a prerequisite for regulation and public trust. Yet major challenges remain. One proposed solution is scalable formal verification. While formal methods offer the strongest guarantees, they are not yet systematically applicable to large neural networks, and current certified robustness techniques yield limited guarantees and reduced accuracy. A key research direction is extending formal verification to hybrid AI via compositional methods, abstraction refinement, and architectures designed for verifiability.

Another important direction is specifying OD for high-dimensional, unstructured inputs. For systems using natural language, raw sensor data or visual scenes, precisely defining the operational design domain is difficult. There is an urgent need for formal OD specification languages for such inputs, and for automated tools for OD boundary detection and coverage measurement.

GenAI safety verification remains largely unsolved. Hallucination and prompt injection in large language models lack systematic verification methods. Manual red-teaming is costly, incomplete and non-reproducible. Automatic red-teaming, formal behavioural specifications for LLMs and rigorous GenAI safety evaluation approaches are open problems.

Current reliability and safety assessments mainly use statistics to measure what a system does, not why. Causal models could reveal root causes, predict failure conditions and greatly strengthen safety cases. Integrating causal reasoning with machine learning is an active research area with important implications for assurance.

Our approach defines multiple assessment dimensions, but combining them into a single assurance claim needs formal methods for aggregating heterogeneous evidence. Research on evidence combination, uncertainty propagation in safety cases, and formal assurance case logics is required.

VI. Conclusion and Future Works

This paper presents the first paradigm-aware approach to evaluating robustness, reliability, and safety of AI systems across data-driven, symbolic, hybrid, and generative AI. It offers: precise operational definitions of these properties; a systematic analysis of how they appear and fail in each paradigm; a survey of assessment methods and evidence types for gap analysis and evidence planning; and a regulatory mapping to the EU AI Act, EASA, DO-178C/ARP 4754A, IEC 61508, and the Confiance.ai end-to-end methodology.

Three conclusions follow. First, assessment must be paradigm-aware: methods for symbolic solvers, deep neural networks, and large generative language models differ, and a uniform approach leaves gaps. Second, robustness, reliability, and safety are interdependent but distinct; each requires separate assessment before integration into a safety case. Conflation creates blind spots: a system reliable in testing may fail under distribution shift, and one robust to perturbations may still be hazardous in some contexts. Third, assurance is a lifecycle activity, not a one-off certification: changing environments, performance drift, and new adversarial techniques demand continual evidence collection, evaluation, and updating. As AI becomes more capable and embedded in critical infrastructure, inadequate assurance becomes more costly. The proposed approach is a step toward an AI assurance engineering discipline that evolves with the technology it governs.

Future work aims to complete the approach defined in [3] going beyond data-driven AI by extending AI engineering methods and tools to symbolic, hybrid, and generative AI.

References

[1] ALTAI, "Assessment list for trustworthy artificial intelligence (altai)", High-Level Expert Group on Artificial Intelligence, European Commission, Tech. Rep., 2019.

[2] A. Awadid et al., "AI systems trustworthiness assessment: State of the art", in *Workshop on Model-based System Engineering and AI, 12th International Conference on Model-Based Software and Systems Engineering (Modelsward)*, 2024.

[3] K. Quintero et al., "An end-to-end method for operationalizing trustwothiness in AI-based critical systems", in *15th Int. Conf. on Performance, Safety and Robustness in Complex Systems and Applications*, 2025.

[4] European Union Aviation Safety Agency, "Artificial Intelligence Roadmap: A Human-Centric Approach to AI in Aviation", Tech. Rep., 2023.

[5] J. Mattioli et al., "AI engineering to deploy reliable AI in industry", in *2023 Fifth International Conference on Transdisciplinary AI (TransAI)*, IEEE, 2023, pp. 228–231.

[6] F. Kaakai and P. Raffi, "Towards multi-timescale online monitoring of AI models: Principles and preliminary results", in *SafeAI, AAAI's Workshop on Artificial Intelligence Safety*, vol. 3381, 2023.

[7] M. Gonzalez et al., "Introducing RUM: A Methodological Contribution for Engineering Trustworthy AI Components in Industrial Systems", in *Proceedings of the AAAI Symposium Series*, AAAI, vol. 7, 2025, pp. 153–160.

[8] X. Li et al., "From features engineering to scenarios engineering for trustworthy AI: I&I, C&C, and V&V", *IEEE Intelligent Systems*, vol. 37, no. 4, pp. 18–26, 2022.

[9] K. Kapusta et al., "Protecting ownership rights of ml models using watermarking in the light of adversarial attacks", *AI and Ethics*, vol. 4, no. 1, pp. 95–103, 2024.

[10] G. Soudain, "EASA Artificial Intelligence (AI) Concept Paper Issue 2: Guidance for Level 1&2 machine learning applications - Issue 2", European Union Aviation Safety Agency, Tech. Rep., 2024.

[11] J. Mattioli et al., "An overview of key trustworthiness attributes and KPIs for trusted ML-based systems engineering", *AI and Ethics*, vol. 4, no. 1, pp. 15–25, 2024.

[12] J. Machrouh et al., "Qualification/validation of AI-augmented ATM solutions for sustainable aviation", *Towards Sustainable Aviation Summit*, 2025.

Defining a Minimal Set of Trustworthy Properties for Reliable Knowledge-Based Systems

Florence de Grancey* Gaëlle Lortal† Claire Laudy†
Amandine Audouy* Florent Chenevier* Joshua Salort*
*Thales, France
†cortAIx Labs, Thales, France
†email:firstname.lastname@thalesgroup.com

***Abstract*—Knowledge-Based Components (KBCs) are a promising technology for a high-risk Artificial Intelligence (AI) system. Nevertheless, ensuring their reliability becomes challenging as Knowledge Models and reasoning mechanisms grow in complexity. Existing trustworthy-AI frameworks predominantly focus on learning systems and offer limited guidance for symbolic of knowledge driven systems. As an initial step toward filling this gap, we propose a minimal set of trustworthy properties specifically tailored to KBCs. These properties are derived from a state-of-the-art analysis, a risk-based examination of potential failure modes, and an adaptation of established trustworthiness principles from the learning domain. We analyze how these properties influence the development process of KBCs and illustrate their practical relevance through a supportive example.**

Keywords-Artificial Intelligence; Knowledge Processing; Knowledge Representation Formalisms and Methods; Knowledge base verification.

I. Introduction

While Artificial Intelligence (AI) is now widely deployed for every-day life applications, its integration into high risk systems, such as nuclear plants or aircraft, remains a major challenge. Such systems are characterized by the possibility of fatal consequences in case of failure, including human injury or environmental damage. They are therefore developed under a stringent and reliable development process and strong regulatory constraints. In the context of AI, *trustworthy AI engineering* practices are essential to guarantee AI system performance, safety and robustness, and its compliance to emerging regulation, such as the AI Act [1]. Theses practices are build upon established classical engineering methods practices (such requirement writing and verification, configuration management) ensuring system reliability, and extended through the definition and verification of *trustworthy properties*. Theses properties may apply in the scope of the AI system or in the scope of items, such as datasets or trained models in the case of supervised learning technologies.

To address the challenge of AI based high-risk applications, the industry is also exploring alternative AI technologies, such as *Symbolic AI* (SAI). Symbolic approaches rely on the formalization of knowledge and facts (general, domain-specific, expert) using logics, and a combination with reasoning algorithm to infer new knowledge. This family includes expert-systems, ontology-based reasoning and constraint problem solving. SAI offers attractive features for high-risk systems, such as explainability by design and reasoning algorithm correctness.

Nevertheless, when applications grows in complexity, requiring large Knowledge Models and complex reasoning algorithms, AI engineer may no longer be able to anticipate the system outcome. Developers may be unable to design the exhaustive required test suite covering the combinatorial explosion of possible facts, knowledge, and reasoning situations. In this context, identification of trustworthy properties for KBC becomes a promising approach to ensure reliability.

Extending the methodological works on ontology building [2] in a water resources monitoring project [1], our contribution provides a starting point toward defining a minimal set of trustworthy properties for SAI System. Building on the state of the art in trustworthy properties for AI, completed by a risk based approach, we identified and defined a preliminary set of properties. Their impact on engineering process is also analyzed. The paper describes the analysis performed to identify trustworthy properties in section 2. Section 3 describes the selected set of properties and their applicability in the context of knowledge base system development. Section 4 presents an illustrative example of theses properties. A discussion is provided in the last section.

II. Defining trustworthy properties

A. Knowledge based component

To support our analysis, we first define the scope under consideration. We consider an AI-based system that incorporates a software component developed using SAI technology, referred as the *Knowledge Based Component* (KBC). A KBC can generally be structured as a combination of several elements, as illustrated in Figure 1.

- A Knowledge Model, which contains the formalized knowledge. It includes general concepts (e.g., "cars", "road",..) and contextual information (e.g., "N124 is a road","N124 is open"). The knowledge is expressed in a dedicated knowledge representation language, defining the allowed syntactic and semantic rules.
- A reasoning engine, which applies a reasoning algorithm to infer new knowledge from the existing knowledge model.

[1]This research was funded, by the Agence Nationale de la Recherche (ANR) through TETRA Project, grant ANR- 22-FAI2-0006-01

Algorithm rely on a set of logical rules, iteratively applied to each element of the knowledge model.
- A Request engine (or query engine), which enables interaction with the knowledge model, by retrieving relevant information. It usually relies on specific search algorithms designed for efficiency.

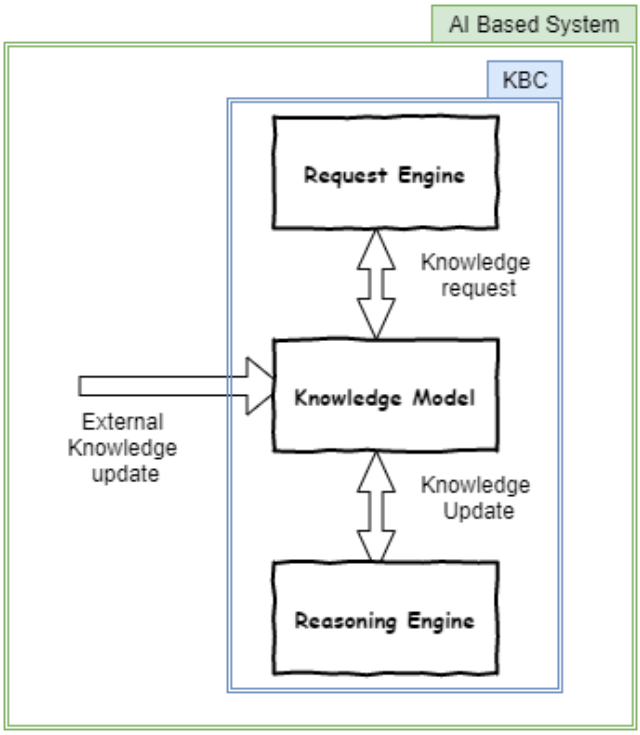


Figure 1. Knowledge Based Component schematic architecture.

It is important to note that the functional behavior emerges from the KBC as a whole, rather than individual elements: A knowledge model alone is not usable without a query mechanism to access its content. Although, this decomposition is inspired by expert systems, it can be generalized to other SAI technologies such constraint-solving problems. In this case, the set of variable, domains, and constraint constitute the knowledge model, while the constraint solver acts as the reasoning engine.

B. Methodology

To define the trustworthy properties for reliable KBC, we apply an exploration methodology structured along three complementary paths:
- State of the art analysis, focusing on trustworthy properties relevant to the context of knowledge modeling and symbolic reasoning.
- Bottom-Up analysis starting from potential errors and risks associated with KBCs and identifying the trustworthy properties required to mitigate them.
- A top-down approach, examining trustworthy properties established in the machine-learning domain and assessing their applicability to KBCs.

C. Path 1: State of the art analysis

Knowledge-Base engineering has provided standards[2], methodologies [3], [4], [5], [6], [7], and tools [8], [9], to build, use and maintain KB and thus provide confidence in KBS. Recent works are more often focused on specific technologies.

Ontology engineering One of the most influential contributions comes from the 2013 Ontology Summit [10], which proposed an eight-step iterative development cycle. This development cycle is driven by five properties: *Intelligibility* ("Can humans understand the ontology correctly?"), *fidelity* ("Does the ontology accurately represent its domain?"), *Craftsmanship* ("Is the ontology well-built and are design decisions followed consistently?"), *Fitness* ("Does the representation of the domain fit the requirements for its intended use?) and *Deployability* ("Does the deployed ontology meet the requirements of the information system of which it is part?"). Additional designed-oriented properties are described in [11] including *Clarity*, *Coherence*, *Extendibility* ("An ontology should be designed to anticipate the uses of the shared vocabulary."), *Minimal encoding bias*, *Minimal ontological commitment*. While theses properties capture relevant desirable qualities for a KBCs, they lack clear definitions and associated metrics or measurement methods.

Knowledge graphs: Knowledge Graphs (KGs) are widely used to store, represent, and manage structured information about entities and their relation. Assessing and evaluating their quality has become an active research axes. [12] provided a comprehensive review of quality requirements in the context of KGs. [13] address the challenge by proposing a set of properties and tractable metrics, such as *"Accuracy and Correctness to measure the degree to which the KG reflects ground truth"* or *"Completeness measures how thoroughly the Knowledge Graph covers its intended domain"*. Generalization of theses metrics to KBCs can be considered;

Neuro-Symbolic AI: Neuro-Symbolic AI (NSAI) is an active research domain aiming at integrating symbolic reasoning and deep-learning, to leverage qualities of both technologies. As detailed in this comprehensive survey, [14], NSAI enforces by design interpretability, robustness and fairness, which are desirable properties for a learning system. However, [15] pinpoint that the development of trustworthiness methods is largely driven by application needs and lack of a general framework. The work of [16] assesses the application of AI Act desirable features for an AI System (e.g., *Human oversight*, *robustness*, *fairness*), in the context of NSAI and shows that a plurality of metrics is still missing to perform the complete evaluation.

D. Path 2: A Risk-Based Analysis

To ensure the reliability of a KBC deployed in high-risk system, the development process must minimize the occurrence of runtime errors. Such errors may affect the system in two ways:

[2]https://www.w3.org/standards/ Accessed: Mar. 5, 2026.

- **Reduced availability**: the component fails to deliver its intended function due to a detected error or a component loss.
- **Reduced integrity**: the component provides an erroneous output, that remains undetected.

Ensuring that a set of trustworthy properties is hold throughout development and operation is a promising approach to mitigation. So, we conduct a systematic analysis of KBC and its components potential errors, and linked to a specific risk. We then derive the adequate trustworthy property required for mitigation.

Knowledge Representation Language (KRL): We assume that the KRL is selected at design time and remains fixed during KBC operations. Errors occur exclusively at design-time and may take the following forms:

- The choice of an over-expressive KRL: High expressivity language can support complex constructs that simplify the modeling (such as the "existential" quantifier). However, it may lead to undecidability (e.g., rule application does not guarantee termination) or to prohibitive computational costs (e.g., exponential time reasoning). Such design choice can lead to an *availability* risk.
- Conversely, selecting a KRL with insufficient expressive power can lead to oversimplification of the Knowledge Model and a potential *integrity* risk.

Mitigation by property 1: We can define a **Language Suitability** property, defined as "the extend to which the knowledge representation language contains only syntactic elements or rules required for considered use cases". This property is related to *Fitness* described in [10].

Knowledge modeling: Knowledge is formalized through the association of general concepts ("car", "road") and instantiated in instance-level elements representing real-world entities ("TN-822" as car number). Errors in knowledge modeling mainly arises from semantically incorrect content, such as false knowledge (“cars can fly”) or inconsistent knowledge (both “all cars have only four wheels” and “all cars have only two wheels”). Such errors directly lead to an *integrity risk*.
Mitigation by property 2: We define a **knowledge correctness** property as the semantic accuracy of the knowledge contained in the KBC. It is worthy to note that it can be refined into a *knowledge consistency* e.g., the Knowledge Model does not contain contradictory elements.

Even when semantic correctness is ensured, additional modeling errors may occur:

- Missing knowledge: required elements lack for executing a use case. It leads to errors and *availability* risks.
- Excess knowledge: irrelevant elements with respect to the UC, increasing the size and complexity. It degrades reasoning performance and leads to *availability* risks.

Mitigation by property 3: We define a **Knowledge Suitability property**, i.e. "the Knowledge Model contains exactly the necessary knowledge element for the use-case". "Necessary knowledge elements" are used at runtime.

Reasoning Algorithm: Reasoning algorithms infer new facts from the existing Knowledge Model. They are evaluated by their *soundness* (all the derived inference are true from semantic point of view), their *completeness* (an algorithm is complete if, when deriving formulas, all the formulas are well derived), and their *algorithmic complexity*. Theses properties allow to identify errors related to:

- Non-sound algorithm e.g., some derived inference may not be true, leading to *integrity risk*.
- Incomplete algorithm e.g., some inference are not computed, leading to erroneous output and *integrity risk*.
- Algorithm that has no termination in a reasonable time for some Knowledge Models, leading to *availability risk*.

Mitigation by property 4: We define the **Algorithm suitability** property, stating "that the reasoning algorithm and its constraints (response times, semantic accuracy of responses) are appropriate for the selected use case". This property is close to *Craftsmanship* in [10].

Request Algorithm: They exhibit similar properties, errors and risk as an information search algorithm, e.g., recall completeness (all requested elements are provided), precision (too much elements are provided) or non-acceptable termination time.

Mitigation by property 5: We extend the suitability concept to define a **Request Algorithm suitability** ensuring that "the request mechanism is adapted to the use case and its performance constraints".

E. Path 3: Extension of Machine Learning trustworthy properties

The third path to define trustworthy properties consists in examining how properties established for learning-based AI systems can be extended to Knowledge-Based Components. Trustworthy AI properties for machine learning have been extensively studied in the literature, notably in the DEEL project white paper [17] and the confiance.ai report [18]. From these analysis, several categories of properties can be identified:

- Properties related to desirable engineering feature. Such properties are independent of the underlying (AI) technology. Examples are “Auditability”, ”Maintenability”, “Resilience”, “Specifiability” and “Verifiability”. Theses properties can be directly transferred to KBCs development process, without modifications.
- Properties can can be easily adapted to KBCs, assuming the specificity of knowledge based applications. For instance, the **"Data Quality"** defined in [17] as *"the extent to which data are free of defects and possess the desired features"* can be transposed to a Knowledge Quality property, aligned with the notions introduced in the risk-based analysis. Similarly, **Explainability**, defined as *“the extent to which the behavior of a ML model can be made*, can be applied to KBCs by replacing the notion of “ML model” with the knowledge base and its associated reasoning mechanisms. It is close to the explainability concepts explored in [15].

III. Minimal set of trustworthy properties

A. Properties selection

Grounded in the three exploration paths, we select a minimal set of trustworthy properties based on three criteria:

- If the property mitigates a risk when satisfied ?
- If it is possible to verify the property at least once during KBC development process (see related section) ?
- If the selected properties are consistent, non-redundant with another selected property ?

We obtain the set of properties given in Table I. It is worth noting that we define both properties related to the whole component, as it hosts the functional behavior of the component, and properties related to KBC elements, as they drive specific failures.

In addition to this set, we advocate that **KBC explainability** should also be considered. The *Development explainability* could be a useful method to detect unexpected behaviors during development. *Operational explainability*, or explainability for the end-user may be a crucial property for acceptability and trust. In the context of KBCs, explainability is related to the ability to store or reconstruct the inference or query path within the Knowledge Model ([19]).

TABLE I. Proposed minimal set of Trustworthy properties for a KBC (A= Availability risk, I = Integrity risk).

Property	Definition	Mitigate
Language suitability	The extend to which the chosen knowledge representation language contains only the syntactic constructs required for the use case	A,I
Knowledge correctness	The extend to which the Knowledge Model is logically and semantically consistent within the use case needs	I
Knowledge completeness	The extend to which the Knowledge Model contains the necessary knowledge to satisfy use case needs	I
Knowledge suitability	The extend to which the Knowledge Model contains the necessary knowledge to satisfy use case needs	A,I
Algorithm suitability	The extend to which the selected reasoning algorithm fits to the use case needs and its constraints	A,I
KBC performance	The extend to which the selected reasoning algorithm fits to the use case performance needs	A,I
KBC stability	The extend to which the KBC fits to the use case stability constraints	A,I
KBC robustness	The extend to which the KBC is able to maintain its behavior in adverse conditions	A,I

B. Verifiability and impact on the engineering process

The defined Trustworthy properties can be smoothly integrated into the development process of a system containing a Knowledge Base Component, as illustrated in Figure 2.

This development cycle begins with traditional system-engineering activities, including the elicitation of end-user needs, the definition of Concepts of Operation, and the specification of system requirements and architecture. A particular attention should be paid to define a complete and representative set of operational scenarios describing how the system will be used. These scenarios will later guide the construction of test cases at the KBC level.

Once responsibilities and requirements have been allocated to the KBC, the proper development of the KBC begins. Following the approach proposed in the confiance.ai project [20], we recommend to consider a component development cycle divided into two main steps:

- A functional Design and Verification/Validation step, which aims to develop the KBC and its internal elements according to the allocated requirements, then to verify the correctness of its functional behavior. For clarity, we distinguish a **functional design phase** from a **functional (validation and) verification** phase.
- An implementation step, where the designed KBC is concretely implemented in the appropriate software stack (e.g., programming language, operating system) and in the appropriate hardware platform (e.g., CPU, FPGA).

We now detail how trustworthy properties are integrated into theses development steps.

Functional design phase: This phase includes activities related to knowledge modeling, selection of the appropriate knowledge-representation language, and development of the reasoning and querying algorithm. Theses activities are typically carried out iteratively to converge toward the most suitable design choice. Trustworthy properties can be introduced at this stage as desirable qualities, guiding the design or expressed directly as requirements. For example, *language suitability* may be used as a design constraint and lead to a requirement, such as *The Knowledge Model shall be expressed in OWL-EL profile* for an ontology-based application. The property *knowledge correctness*, may lead to define supplementary requirements related to runtime validation rules, such as SHACL rules [21] for an ontology-based application. The property *algorithm suitability* may motivated requirements related to the soundness and completeness of the selected reasoning or querying algorithms.

During knowledge modeling, we encourage to set-up a **bidirectional traceability** between the operational scenarios defined at system level and knowledge elements contained in Knowledge Model. It can be an effective method to ensure *knowledge completeness* and *knowledge suitability* during the design.

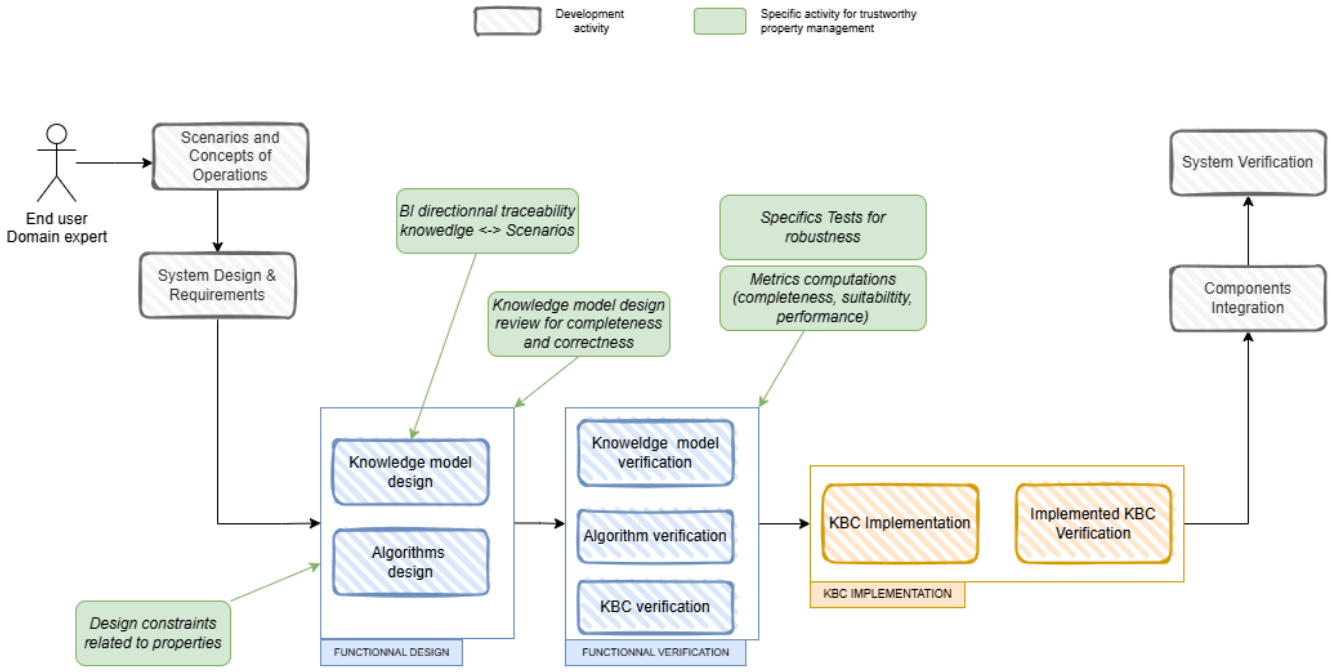


Figure 2. (Simplified) Knowledge Based Component development process.

At the end of the design process, we also recommend to conduct **knowledge-model design review** that can be performed to assess properties knowledge related properties. This review can be assisted by tools to compute metrics on the Knowledge Model, such as OntoMetrics in the ontology context ([22]). Custom metrics may be defined, such as syntactic construct coverage metric as the number of syntactic construct used over the total number of syntactic constructs of the KRL, for *language suitability*.

Functional verification of the KBC: This step focuses on validating and verifying the designed KBC with respect to its requirements and trustworthy properties. The usual practice is to build tests procedures and tests cases that exercise the KBC and its elements under various querying and reasoning conditions. The set of operational scenarios identified during system-level activities may be used as guidance during their elaboration. Several trustworthy properties can be assessed through this scenario-based evaluation: *Algorithm suitability* can be verified using measurement of computational load and the accuracy of the KBC outcome; *Knowledge suitability* can be assessed by computing a knowledge coverage metric defined as the number of knowledge elements used over the total number of knowledge elements. Complementary tests can be developed to address specific trustworthy properties like *KBC robustness*. This property will require evaluating the KBC under particular configurations of knowledge elements or atypical request patterns, that may be tool-assisted [23].

KBC implementation Once the functional behavior of the designed KBC has been validated, the proper **implementation** activities can begin. This phase consists of implementing the KBC on the appropriate software stack and/or hardware platform. To the best of our knowledge, traditional implementation activities like software development are directly applicable. After implementation, compliance with the trustworthy properties should then be re-verified on the implemented KBC, to guarantee that no degradations occurs during the process. Only after this verification, the implemented KBC be integrated into the broader system-engineering workflow. The operational scenarios defined during system design can then be reused to confirm that the final system continues to meet end-user needs.

C. Use case illustration

We illustrate the proposed trustworthy properties using a simple supportive example. Consider a use case in the automotive domain, where the system assists the driver by answering drivers questions, such as *if their car can drive on a specific road?*. A system embedding Knowledge Based component is used to compute reliable answers.

The designed Knowledge Model (KM) is represented in Figure 3. We adopt the following notation: in square brackets or blue in the figure, the general concepts (ex: [car]), in brackets or orange in the figure, the facts (ex: (TN-822) as identifier of the driver's car). The KM also contains a simple deductive rule R1:

```
(R1) A [Car] CanDriveOn [Road] if [Road]
isTypeOf (Highway) or (Back road)
```

We define two operational scenario related to queries (Q):

```
Q1: Is (TN-822) CanDriveOn (A64) ?
Expected Outcome: Yes
```

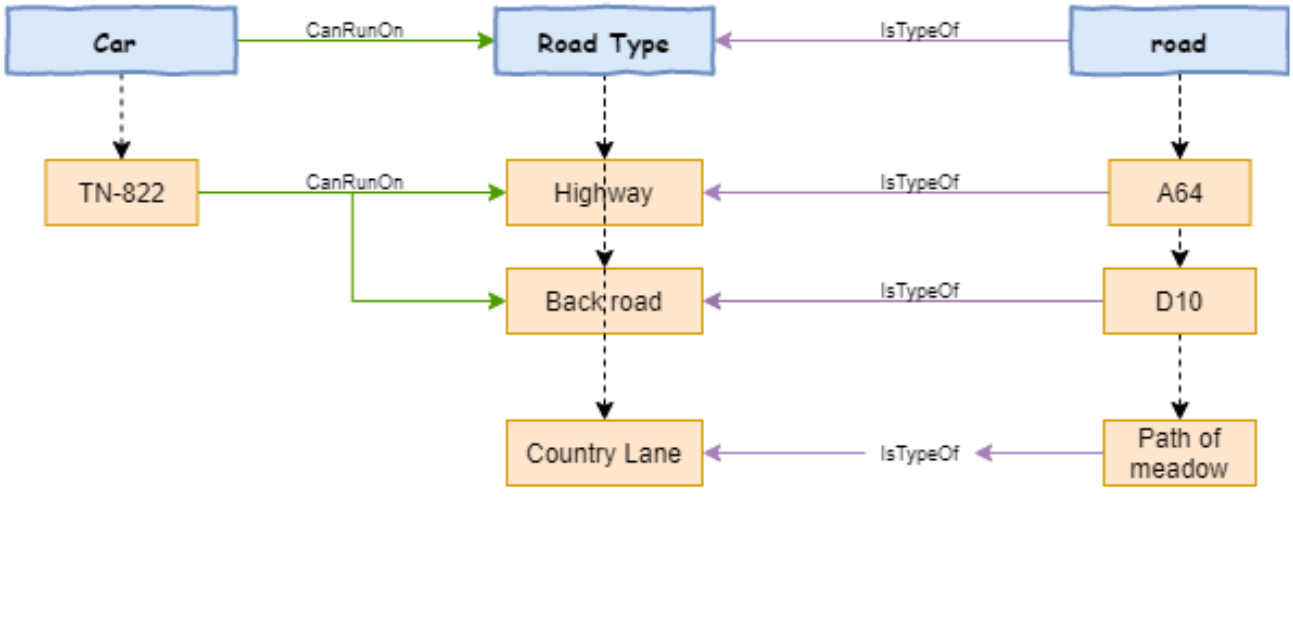


Figure 3. Automotive Use Case Knowledge Model. Concepts are represented in blue rectangles, facts in oranges rectangles. Relations between concepts are arrows with text.

```
-----
Q2: On which [road] (TN-822)
CanDriveOn ?
Expected Outcome: A64,D10
```

Is it worth noting that the KM does not explicitly express the expected outcomes in both scenarios. The answers are obtained by applying the deductive rule to the KM.

We now illustrate potential errors are mitigated by trustworthy properties:

Knowledge correctness: This property is violated if an element is incorrectly specified in the KM. For example, if the road (A64) is assigned an incorrect type, the query Q1 produces an erroneous outcome:

```
KM error: "(A64) isTypeOf (CountryLane)"
Q1: Is (TN-822) CanDriveOn (A64) ?
Outcome: No
```

Knowledge completeness: This property is violated if a necessary knowledge element is missing. In such a case, the KBC cannot produce the correct answer:

```
KM error: Missing Element
"(A64) isTypeOf (Highway)"
Q2: On which [road] (TN-822) CanDriveOn ?
Expected Outcome: D10
```

Knowledge suitability: The property will be violated if the KM contains concepts of [bike] "canRunOn" [CountryLane], which is not necessary for the use case. The outcome is still true but the reasoning time can be increased.

```
Knowledge base additionnal element:
[bike]  "canRunOn" [CountryLane]
Q1:  Is (TN-822) CanDriveOn (A64) ?
Outcome: Yes
```

Algorithm suitability: The property is not satisfied in this use case if the selected algorithm is unable to apply the deductive rule. In such a situation, the KBC can no longer produce the expected result.

KBC stability: Stability can be assessed by testing different formalization of the request, such as expressed below. A stable KBC will answer yes regardless the formalization.

```
Q1a: Is my (TN-822) CanDriveOn (A64) ?
Q1b: Is (TN-822) CanDriveOn (A64)
or not?
Expected Outcome: Yes
```

KBC robustness: On the toy use case, KBC robustness can be experimented by assuming errors on the query, such as:

```
Q3a: Is (TN-821) CanDriveOn (A64) ?
Expected Outcome: No
Q3b: Is (TN-822) CanDriveOn (D11) ?
Expected Outcome: No
```

IV. Discussion

Based on the state-of-the art analysis and the risk-based approach, we propose a minimal set of trustworthy properties for KBCs. We acknowledge that this minimal set was developed from an engineering-oriented perspective and could be enriched with additional properties addressing human-factors or ethical considerations. For example, particular attention should be paid to undesirable biases, which may easily emerge during knowledge modeling and lead to unappropriate use of the system. A property of **Knowledge fairness** may be defined as Knowledge Model level to mitigate this ethical risk, and verified during the knowledge design review.

A mandatory criterion for selecting the properties in this minimal set was the ability to **verify** them during development process. We do not prescribe any specific verification methodology - design review, test, traceability analysis,... -, however, we pinpoint that a formal demonstration of each property is currently out of reach due to the absence of a sufficiently mature mathematical framework. We therefore encourage the academic community to pursue the formalization of these properties and the development of associated verification methods. This would constitute a valuable contribution to improving the reliability of knowledge-based applications for high-risk systems.

The relevance of trustworthy properties approach may be challenged by considerations related to use case complexity. In simple rule-based systems, where the behavior can be fully specified and the reasoning/query space can be exhaustively validated using a manageable set of test scenarios, such properties may appear unnecessary. In these situations, traditional verification techniques are often sufficient to guaranty the reliability of the KBC. We argue that trustworthy properties become essential once the system exceeds a certain **complexity threshold**. This threshold can be characterized by several criteria, including:

- the domain expert's ability to handle and understand the KM (without being a knowledge modeling expert).
- the number of reasoning steps required to answer user needs.
- the developer's ability to build the adequate and exhaustive set of scenarios exercising the reasoning and query over the KBC,
- the domain expert's ability to anticipate the KBC's outcomes on these scenarios.

Such criteria should be refined by application on uses-cases with different levels of complexity.

V. Conclusion and future work

In this paper, we proposed a preliminary set of trustworthy properties for Knowledge-Based Components. These properties were identified through an analysis of consensus-based trustworthy-AI principles and a risk-based approach, and then specialized for the context of symbolic-AI systems. We acknowledge that these properties may be excessive for simple KBCs, where a requirement-based approach may suffice, but they become necessary for more complex applications.

Our approach is grounded in engineering practices for AI development. To advance this work, we encourage the research community to further refine the definition of trustworthy properties and, in particular, to develop their mathematical formalization and associated verification methods. Such contributions would significantly strengthen the reliability and trustworthiness of knowledge-based systems.

Our future work focuses on implementing verification methods associated with the trustworthy properties identified in this study. We are developing software tools to support automated verification, such as metric-based assessments—as well as methodologies to guide design reviews of Knowledge Models. These methods and tools are being exercised on several industrial use cases exhibiting different levels of complexity.

References

[1] Coll., *Ai act*, https://artificialintelligenceact.eu/, 2022. Accessed: Mar. 5, 2026.

[2] M. Zenner et al., "Tetra–from methodology to operational tools for water-based ai projects", Copernicus Meetings, Tech. Rep., 2026.

[3] G. Schreiber, B. Wielinga, and J. Breuker, *KADS: A principled approach to knowledge-based system development*. Academic Press, 1993, vol. 11.

[4] G. Schreiber, B. Wielinga, W. Jansweijer, et al., "The kactus view on the 'o'word", in *IJCAI workshop on basic ontological issues in knowledge sharing*, vol. 8145, 1995.

[5] M. Uschold et al., "Building ontologies: Towards a unified methodology", *Technical report-university of Edinburgh artificial intelligence applications institute AIAI TR*, 1996.

[6] M. Fernandez, A. Gomez-Perez, and M. Juristo N, "From ontological art towards ontological engineering", in *Proceedings of the Spring Symposium Series on Ontological Engineering (AAAI'97), AAAI Press*, 1997.

[7] W. Ceusters, "Towards a realim-based metric for quality assurance in ontology matching", in *Formal Ontology in Information Systems: Proceedings of the Fourth International Conference (FOIS 2006)*, B. Bennett and C. Fellbaum, Eds., IOS Press, vol. 150, 2006, p. 321.

[8] J. Blázquez, M. Fernández, J. García-Pinar, and A. Gómez-Pérez, "Building ontologies at the knowledge level using the ontology design environment", in *Proceedings of the 11th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW'98)*, vol. 2, University of Calgary, Alberta, Canadá, Apr. 1998. Accessed: Mar. 5, 2026. [Online]. Available: https://oa.upm.es/6457/.

[9] M. Fernández López and A. Gómez-Pérez, "The integration of ontoclean in webode", in *CEUR Workshop Proceedings*, 2002.

[10] A. Vizedom et al., "Toward ontology evaluation across the lifecycle", *Applied ontology*, vol. 8, pp. 179–194, Oct. 2013. DOI: 10.3233/AO-130125.

[11] T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing?", *International Journal of Human-Computer Studies*, vol. 43, no. 5, pp. 907–928, 1995, ISSN: 1071-5819. DOI: https://doi.org/10.1006/ijhc.1995.1081. Accessed: Mar. 5, 2026. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1071581985710816.

[12] B. Xue and L. Zou, "Knowledge graph quality management: A comprehensive survey", *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 5, pp. 4969–4988, 2023. DOI: 10.1109/TKDE.2022.3150080.

[13] J. Mattioli, L. Mattioli, and M. Gonzalez, "A brief overview of key quality metrics for knowledge graph solution illustration on digital notams", *Proceedings of the AAAI Symposium Series*, vol. 7, pp. 206–213, Nov. 2025. DOI: 10.1609/aaaiss.v7i1.36888.

[14] J. Pittman, L. Eddy, and K. Wiseman, "Responsible reasoning - a systematic review", *Preprints*, Oct. 2024. DOI: 10.20944/preprints202410.0985.v1. Accessed: Mar. 5, 2026. [Online]. Available: https://doi.org/10.20944/preprints202410.0985.v1.

[15] C. Michel–Delétie and M. K. Sarker, "Neuro-symbolic methods for trustworthy ai: A systematic review with a focus on interpretability", *Neurosymbolic Artificial Intelligence 0*, 2024.

[16] A. Agiollo and A. Omicini, "Measuring trustworthiness in neuro-symbolic integration", in *2023 18th Conference on Computer Science and Intelligence Systems (FedCSIS)*, 2023, pp. 1–10. DOI: 10.15439/2023F6019.

[17] F. Mamalet et al., "White paper machine learning in certified systems", IRT Saint Exupéry ; ANITI, Research Report, Mar. 2021. [Online]. Available: https://hal.science/hal-03176080.

[18] J. Mattioli et al., "An overview of key trustworthiness attributes and kpis for trusted ml-based systems engineering", *AI and Ethics*, vol. 4, no. 1, pp. 15–25, 2024.

[19] Y. Ye, X. Cui, and D. Ouyang, "Extracting a justification for owl ontologies by critical axioms", *Frontiers of Computer Science*, vol. 14, no. 4, p. 144 305, 2020.

[20] C. Project, *Confiance.ai body of knowledge*, https://bok.confiance.ai/. Accessed: Mar. 5, 2026.

[21] R. David, D. Habgood, A. Seaborne, and S. Steyskal, *W3c : Shacl12 rules*, https://www.w3.org/TR/shacl12-rules/. Accessed: Mar. 5, 2026.

[22] B. Lantow, "Ontometrics: Putting metrics into use for ontology evaluation.", in *KEOD*, 2016, pp. 186–191.

[23] C. Laudy and N. Museux, "Peacock: A benchmarks generation framework for high-level information fusion evaluation", in *Fusion 21*, Nov. 2021. DOI: 10.23919/FUSION49465.2021.9627038.

What IF: Ultimate Intelligence FORGIVES?

Sharron Framminghamm
The Human Engine
Angers, France
e-mail: sframmingam@outlook.com

Abstract—**The future of work depends not just on technical development but on the quality of human decision-making, values and behaviors that shape this development. While Artificial Intelligence development continues to excel, its long-term safety and sustainability is strengthened through human-in-the-loop (HITL) systems. Forgiveness, often overlooked, serves as a foundational component of advanced decision-making. Unlike Artificial Intelligence, humans possess the capacity for humility, restraint, instinct and reconciliation, qualities that enable forgiveness and are vital for ethical Artificial Intelligence governance. This research introduces the FORGIVES framework, a practical model translating high-level ethical principles (like the Asilomar Artificial Intelligence Principles) into actionable human-centered behaviors. This framework aligns Artificial Intelligence with human shared values, promoting transparency, responsibility and long-term benefits while addressing global challenges in operationalising shared ethics. Ultimately, forgiveness is a strategic necessity. Therefore, improving human capacity for forgiveness is crucial for decision-making that develops Artificial Intelligence systems that are adaptive, restorative and aligned with human well-being. The distinction between human and artificial intelligence, lies in human moral judgment and relational intelligence, qualities that must guide Artificial Intelligence's evolution to serve the greater good and increase human performance. The forgiveness framework can be understood through endurance environments. Endurance performance reveals how human beings respond under pressure, fatigue, failure, uncertainty and prolonged challenge. These factors parallel the development and industry of Artificial Intelligence. To develop decision-making abilities, it is beneficial to seek and not avoid endurance environments, as they aid the understanding of something increasingly relevant in the age of Artificial Intelligence - the difference between optimisation and wisdom.**

Keywords-Artificial-Intelligence; decision-making; forgiveness; Human-in-the-Loop; endurance.

I. INTRODUCTION

As Artificial Intelligence systems are increasingly evaluated on performance, safety and robustness, a critical but underexplored gap is emerging: not a shortage of technical capability, but a shortage of high-quality human decision-making. This research argues that the long-term sustainability and responsible development of Artificial Intelligence systems depends not only on engineering excellence, but on the character, judgment and behavioral frameworks of those designing, deploying and interacting with them.

Lazaros, Vrahatis & Kotsiantis [1] highlights that Human-in-the-Loop (HITL) Artificial Intelligence has emerged as a significant direction within the field, particularly in environments where fully automated systems are unable to adequately account for context, ethical judgment, ambiguity, or accountability. While Artificial Intelligence systems are increasingly capable of processing vast quantities of information and identifying complex patterns, they remain limited in their ability to interpret human nuance, social consequence and moral responsibility in dynamic real-world situations.

Human-in-the-Loop approaches address this limitation by ensuring meaningful human participation within critical stages of Artificial Intelligence development, deployment, monitoring and decision validation. Rather than positioning humans as passive overseers, Human-in-the-Loop frameworks recognise human judgment as an active and necessary component of safe and effective Artificial Intelligence operation. This is particularly important in safety-critical and socially sensitive domains such as healthcare, transportation, legal systems, education and emotionally responsive Artificial Intelligence applications, where decisions may carry significant human impact.

Amershi et al. [2] provide practical guidance for designing Artificial Intelligence systems that support effective human interaction and oversight and demonstrate that successful Artificial Intelligence deployment depends heavily on human-centered system design rather than technical performance alone. Human-in-the-Loop systems are widely becoming accepted as essential - but effectiveness depends on the decision-making skills of the human in the loop. This research extends existing Human-in-the-Loop approaches by arguing that not all human input is equal.

Additionally, this research builds on the work of Russell and Norvig [3], who emphasise Value Alignment and that advanced Artificial Intelligence systems must remain aligned with human goals and subject to meaningful human oversight. Their work highlights the limitations of purely optimisation-driven systems and the importance of uncertainty, value alignment and corrigibility in Artificial Intelligence behaviour. This research addresses the challenge they raise of defining and operationalising shared values across global contexts. This challenge is addressed by drawing on historical analysis. This research traces the influence of early legal and moral codes - originating in texts such as Exodus [4] and later teachings expanded in Matthew

[5], on the development of global legal systems, including principles of justice, accountability, human dignity and mercy. These foundations provide a tested model for encoding values into systems that govern behaviour at scale. In Matthew [6] we discover 8 global values shared across humanity. Regardless of your faith perspective all humanity share the 8 beatitudes outlined in that message. They flip the script by helping us to understand the value of seeing things differently and helps us to cultivate a mindset that FORGIVES. They also help us to identify human attributes that are crucial for establishing moral and ethical codes alongside the Future of Life Institute Asilomar Artificial Intelligence Principles [7].

This paper also addresses the additional challenge of how to apply global ethical frameworks operationally. This research translates high-level ethical guidelines into a practical, human-centered actionable model for decision-making and behaviour. While existing principles define what responsible Artificial Intelligence should achieve, this research provides a framework to explore how individuals and organisations can embody these principles in practice.

This industry research investigates forgiveness as a foundational yet overlooked component of advanced decision-making. While Artificial Intelligence systems can simulate aspects of emotional intelligence, they cannot experience or embody the conditions that enable forgiveness - such as humility, loss, restraint, moral tension, or reconciliation. This limitation reveals a boundary of artificial intelligence and highlights a critical domain in which human capability remains essential. As described by Dignum [8] 'The more that AI can do, the more it underscores the irreplaceable qualities of human creativity, empathy, and moral reasoning.'

This paper proposes forgiveness as a unifying, cross-cultural principle - widely understood, practically applicable, and scalable across diverse environments. Unlike abstract ethical constructs, forgiveness functions as both a mindset and a behavioural framework, offering a consistent basis for decision-making in complex, real-world systems.

This research uses endurance principles to illustrate forgiveness because endurance correlates well to the long-term development perspective required for Artificial Intelligence safety. Endurance performance also reveals how human beings respond under pressure, fatigue, failure, uncertainty and prolonged challenge. In these moments, decision-making becomes visible. Endurance is not sustained through intensity alone, but through humility, restraint, reflection, perseverance and the ability to recover well after setbacks.

Forgiveness operates in a similar way. It is rarely a single emotional moment; rather, it is an ongoing process of releasing failure, recalibrating perspective and choosing constructive action despite discomfort or disappointment. Like endurance, forgiveness requires disciplined thinking, emotional regulation and long-term vision.

Endurance environments also expose something increasingly relevant in the age of Artificial Intelligence: the difference between optimisation and wisdom. A machine may calculate pace, output, or probability, but human performance depends on qualities such as judgment, meaning, empathy and resilience. These are not simply technical abilities but deeply human capacities that shape how decisions are made over time.

For this reason, endurance principles provide a practical and accessible framework for exploring forgiveness, not only as a moral concept, but as a performance capability that strengthens leadership, Human-in-the-Loop decision-making and responsible Artificial Intelligence development. Through applied analysis of events, cross-sector professional insight and observational patterns from employment and Artificial Intelligence adjacent environments, this research introduces the FORGIVES framework.

II. THE FRAMEWORK

There are 8 separate components of the FORGIVES framework aligned with a transferable human skill that enhances decision-making quality, reduces systemic risk and supports the development of Artificial Intelligence systems that are resilient, ethical and aligned with human well-being. For each component of the framework there are Human-in-the-Loop learnings from a specific case-study of both historical and recent events from within the automotive, aerospace, healthcare, railway and defense industries. Table 1 introduces each component of the framework and provides an overview of each corresponding human attribute, Human-in-the-Loop (HITL) behaviour, Asilomar alignment and endurance principle.

TABLE I. FORGIVENESS FRAMEWORK OVERVIEW

FORGIVES Component & Human Attribute	**Framework**		
	HITL System Behaviour	***Asilomar Alignment***	***Endurance Principle***
Fail: Humility	Transparent error reporting, review cycles, learning loops	Research Goal,	Learn early, fail safely, improve continuously
Originality: Discernment	Diverse human input, context-aware decision review	Responsibility, Failure Transparency	Context matters - data needs human understanding
Respect: Self-Control	Staged deployment, safety gating, human approval checkpoints	Science-Policy Link,	Progress is a process that is planned and paced
Generosity: Integrity	Explainability, audit trails, accountable oversight	Personal Privacy,	Accountable openness aids collaboration
Innocence: Consideration	Ethical data handling, consent-based processes	Liberty and Privacy	Protect positively and apply spherical kindness
Values: Purpose	Human override, escalation pathways, decision validation	Race Avoidance,	Purpose focuses and fuels performance

FORGIVES Component & Human Attribute	Framework		
	HITL System Behaviour	***Asilomar Alignment***	***Endurance Principle***
Empathise: Reconciliation	Safeguarding protocols, escalation for vulnerable users	Safety,	Human care and oversight is critical
Sustain: Commitment	Continuous training, monitoring, recalibration	Risks	To maintain standards commit carefully

A. FORGIVES Framework Component 1: Fail (Humility through rest and reflection)

Cullen [9] described how a train collision near London Paddington station in the UK resulted in fatalities. However, there were warning signs which were missed. Had there been sufficient learning from previous near misses, could the collision have been avoided? It is impossible to know, but it is important to consider what factors prevent learning from previous mistakes.

In high-performance environments, failure is often hidden due to fear, pressure, or reputation. Yet research across safety-critical industries shows that unacknowledged small failures are often the precursors to major incidents. An alternative approach is to treat failure as data.

Historically, this principle is captured in [10] and [11]. Principles across cultures and traditions capture the importance of a structured and scheduled pause, to intentionally step back from activity to review, reflect and reset. Taking a pause reflects humility, recognising limitations and remaining open to correction. These principles encourage intellectual humility and disciplined reflection, as both are essential for learning. When there is an understanding that each person always has something to learn, failure is just a part of life's learning process then it is easier to smile, get back up and go again.

Human-in-the-Loop (HITL) systems and Asilomar alignment application:

- Research Goal – Test, rest and reflection need to be incorporated into initial research goals.
- Responsibility – From the beginning it is important to realise anyone using Artificial Intelligence is shaping what it will become.
- Failure Transparency – This develops a responsible learning culture.

When considering endurance environments Brymer and Oades [12] observed transformation through humility. Together, these principles show that humility and structured reflection are not abstract ideals but practical safeguards; when embedded into Human-in-the-Loop systems through transparent failure reporting and intentional review cycles, they directly support Asilomar principles of responsibility and failure transparency, enabling Artificial Intelligence systems that learn safely rather than scale unchecked error.

B. FORGIVES Framework Component 2: Original Design (Identity, Context & Human Understanding)

Angwin, Larson, Mattu and Kirchner, [13] reported the COMPAS risk assessment tool used in US courts to predict reoffending risk was found to exhibit racial bias, disproportionately flagging Black defendants as higher risk. The system relied on historical data patterns without sufficient contextual understanding of social and structural factors. When systems are designed without recognising the full human context behind data, they risk reinforcing bias and inequity.

Historically, this principle is captured in [14] and [15]. These principles speak, in universal terms, to respect for origin, lived experience and human context. They remind us that people are not neutral data points – each person is unique with individual originality, shaped by history, relationships and circumstance. Ignoring this will lead to incomplete or distorted decision-making. It is easy to drift through life trying to be just like everyone else but each fingerprint proves every person is unique and one of a kind. Humans are not machines, not an exact replica model of each other. When this is understood it can be celebrated and what can be perceived as a weakness can become the greatest asset and most valuable data.

Human-in-the-Loop (HITL) systems and Asilomar alignment application:

- Science-Policy Link: A healthy exchange between people in the process is essential (meaning Artificial Intelligence researchers and policymakers).
- Personal Privacy: Individuals should be aware and control the data they generate.
- Liberty and Privacy: AI should not negatively impact personal freedoms or privacy.

When considering endurance environments Smits, Pepping and Hettinga [16] observed that every endurance performance is different, therefore one-size-fits-all training is not the best approach. By recognising the uniqueness of human experience, these principles reinforce that data cannot be separated from context; when applied through Human-in-the-Loop oversight and diverse input, they align with Asilomar commitments to human values, privacy, and liberty, ensuring Artificial Intelligence systems reflect the richness of real-world humanity rather than narrow historical patterns.

C. FORGIVES Framework Component 3: Restraint (Respect and Patience for Process)

National Transportation Safety Board [17] found that autonomous vehicle testing contributed to a pedestrian death in Arizona, USA. Investigations highlighted gaps in safety oversight, monitoring and escalation processes. While we cannot know if the tragedy could have been avoided, it demonstrates how the pressure for rapid progress can lead to compromised safeguards.

Historically, this principle is captured in [18] and [19]. These reflect the importance of controlled strength, restraint, and respect for the process. A rush to see results can create impatience and frustration that can lead to mistakes. Alternatively, a respect for the process with an understanding

that overnight success does not exist, creates an appreciation for every day and that every single snail step counts.

Human-in-the-Loop (HITL) systems and Asilomar alignment application:

- Race Avoidance – Artificial Intelligence deployment must prioritise safety over speed.
- Safety – Progress must not outrun governance.
- Risk – Staged testing and validation must be enforced.

When considering endurance environments Skorski and Abbiss [20] observed that endurance performance is fundamentally a planned paced process – having patience and respect for the process is fundamental. These principles demonstrate that restraint is a form of strength; when operationalised through Human-in-the-Loop controls, staged deployment and enforced safety protocols, they align directly with Asilomar priorities of safety and race avoidance, ensuring that progress is governed, not rushed.

D. FORGIVES Framework Component 4: Integrity (Transparency & Faithfulness)

House Committee on Transportation and Infrastructure [21] reported that the Boeing 737 MAX crisis demonstrated how design flaws, communication failures and inadequate training contributed to fatal outcomes. At its core, this case reflects a breakdown in transparency and accountability.

Historically, this principle is captured in [22] and [23]. These principles point toward faithfulness, integrity and alignment between intention and action. Survival mindset has nothing to give and avoids transparency. But with generosity with resources (money, time and skills) and an openness to accountability; others are encouraged to do the same, creating trust.

Human-in-the-Loop (HITL) systems and Asilomar alignment application:

- Research Funding – Safety must be prioritised over commercial pressure.
- Human Control – Human accountability must be maintained.
- Judicial Transparency – It is important to ensure decisions are explainable and auditable.

When considering endurance environments Hyland-Monks, Cronin, McNaughton and Marchant [24] observed that high performance endurance becomes sustainable through cognitive accountability. Integrity bridges intention and action; when upheld through Human-in-the-Loop accountability and explainable systems, it supports Asilomar principles of transparency and human control, building trust in Artificial Intelligence systems that are not only effective, but dependable and open to scrutiny.

E. FORGIVES Framework Component 5: Innocent Kindness (Responsibility & Stewardship)

Information Commissioner's Office [25] reported the Google DeepMind / NHS Royal Free case raised concerns about patient data being used without sufficient transparency or informed consent. This highlighted the ethical risks of innovation without accountability.

Historically, this principle is captured in [26] and [27]. These reflect respect for what belongs to others and compassionate responsibility in how we act. When seemingly kind intentions come from a mixed motive it steals authenticity and damages relationships. But when kindness is without motive, it becomes spherical, three dimensional; spherical kindness has consideration for all.

Human-in-the-Loop (HITL) systems and Asilomar alignment application:

- Shared Benefit – Data and Artificial Intelligence should benefit all stakeholders.
- Shared Prosperity – Systems must respect rights and dignity.

When considering endurance environments Thiel, Pfeifer and Sudeck [28] observed that endurance performers must respect their own and others limits. This calls for spherical kindness and consideration. These principles highlight that true responsibility is both ethical and relational; when embedded in Human-in-the-Loop processes that respect consent and dignity, they align with Asilomar principles of shared benefit and shared prosperity, ensuring Artificial Intelligence development serves people rather than exploits them.

F. FORGIVES Framework Component 6: Value Alignment (Truth & Moral Clarity)

Hoffman [29] stated that when Soviet officer Stanislav Petrov chose not to act on a false automated missile alert, this prevented potential catastrophe. This decision demonstrates the importance of human judgment over automated outputs.

Historically, this principle is captured in [30] and [31]. These reflect truth, honest values and clarity of intention. It is easy to go through life chasing what is not important, leaving little energy for what really matters. Alternatively, working every day towards what is truly significant, removes the 9-5 mentality, increases energy, multiplies effort and enhances performance.

Human-in-the-Loop (HITL) systems and Asilomar alignment application:

- Value Alignment – AI must reflect human priorities.
- Human Values – Human judgment must remain central.
- Non-subversion – Systems must not override ethical reasoning.

When considering endurance environments Brick, MacIntyre and Campbell [32] observed that goal-directed thinking regulates endurance performance. Endurance performers have strong, clear and aligned values that propel them forward. Truth and clarity of intention ensure that systems remain grounded in what matters; when reinforced through Human-in-the-Loop judgment and oversight, they directly support Asilomar principles of value alignment, human values and non-subversion, ensuring Artificial Intelligence remains accountable to human priorities.

G. *FORGIVES Framework Component 7: Empathetic Interaction (Relational Awareness & Peace)*

Associated Press. [33] reported a litigation involving Character.AI that highlighted the risks of emotionally intense AI interactions with vulnerable users, including a case linked to a teenager's death. These systems demonstrated the limits of simulated empathy.

Historically, this principle is captured in [34] and [35]. These principles reflect respect for others, relational boundaries and the pursuit of peace. When confronted with conflict it is really easy to take it personally and carry it forward into the next interaction, creating a chain-anger-reaction that can cloud judgement. Alternatively, when an empathetic response is chosen and the other person's perspective is considered, then it is easier to dissolve conflict and create forward momentum.

Human-in-the-Loop (HITL) systems and Asilomar alignment application:

- AI Arms Race – Avoid competitive escalation without safeguards.
- Capability Caution – Recognise limits of emotional Artificial Intelligence.
- Importance – Safeguard vulnerable users through human oversight.

When considering endurance environments McCormick, Meijen, Anstiss and Massey [36] observed that endurance performers that regulate emotion and support others, outperform their technically stronger competitors. These principles remind us that relationships require care, not simulation; when Human-in-the-Loop safeguards are applied to emotionally sensitive interactions, they align with Asilomar principles of capability caution and responsible development, ensuring vulnerable users are protected where Artificial Intelligence alone cannot fully understand.

H. *FORGIVES Framework Component 8: Sustained Performance (Commitment & Endurance)*

National Aeronautics and Space Administration [37] reported repeated aviation incidents. As a result, Crew Resource Management (CRM) was introduced in 1979 in the USA, embedding continuous training, communication discipline and teamwork into safety culture.

Historically, this principle is captured in [34] and [38]. These reflect commitment, discipline and perseverance under pressure. Commitment is not always the best answer; there is a time to commit and a time to quit. When there is full commitment, there is the ability to go the extra mile, become more creative, industrious and effective and find a way around whatever obstacles are faced.

Human-in-the-Loop (HITL) systems and Asilomar alignment application:

- Research Culture – It is important to promote continuous improvement.
- Recursive Self-Improvement – Systems must evolve responsibly.
- Greater Good – Long-term ethical focus must be maintained.

When considering endurance environments Cowden and Worthington [39] observed the cycle of constructive self-forgiveness to maintain and sustain commitment. Sustained excellence is built through disciplined consistency over time; when supported by continuous Human-in-the-Loop training, monitoring, and governance, these principles align with Asilomar commitments to long-term safety and research responsibility, ensuring Artificial Intelligence systems remain reliable, ethical and resilient for all.

Fig 1 shows that the FORGIVES framework is a continuous cycle, that requires daily application and review. It is never a finished process but is a framework for continuous development, improvement and progress.

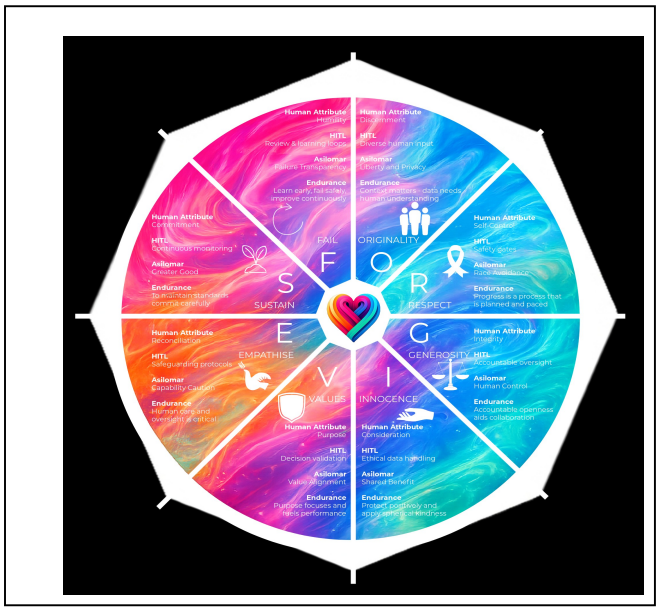


Figure 1. FORGIVES Framework Wheel

III. CONCLUSION AND FUTURE WORKS

This research concludes that the future of Artificial Intelligence will not be determined solely by advances in computational power or model capability, but by the quality of human inputs - decisions, behaviours and values that shape these systems. As Artificial Intelligence is trained on human-generated data and deployed at scale, the distinction between human and artificial intelligence becomes increasingly defined by our capacity for moral judgment and relational intelligence.

Together, these components demonstrate that ancient principles of humility, integrity and responsibility when translated into Human-in-the-Loop practice, aligned with Asilomar governance and applied with endurance principles provide a practical and scalable foundation for building Artificial Intelligence systems that are not only intelligent, but trustworthy.

If decision-making shapes the future of Artificial Intelligence, then improving decision-making is a global imperative. This work positions forgiveness not as a peripheral moral concept, but as a central mechanism for developing intelligence that is adaptive, restorative and sustainable. In this context, becoming a people who

FORGIVES is not only a personal aspiration, but a strategic requirement for shaping intelligent systems that serve the greater good and increase human performance. The forgiveness framework is best understood through the consideration and most pertinently the personal experience of endurance environments. Endurance performance reveals how human beings respond under pressure, fatigue, failure, uncertainty and prolonged challenge. These factors parallel the development and industry of Artificial Intelligence. To develop decision-making abilities, it is beneficial to seek and not avoid endurance environments. As they aid the understanding of something increasingly relevant in the age of Artificial Intelligence - the difference between optimisation and wisdom.

A new company has been formed (The Human Engine) to deliver training and support to organisations to embed the FORGIVES framework into their operations via a training and development program through an endurance environment. Results will be monitored and a future paper is planned.

REFERENCES

[1] K. P. Lazaros, A. G. Vrahatis & S. Kotsiantis, Human-in-the-Loop Artificial Intelligence: A Systematic Review of Concepts, Methods, and Applications. Entropy, 28(4), 377, 2026, https://doi.org/10.3390/e28040377

[2] S. Amershi, et al. “Guidelines for Human-AI Interaction,” Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2019.

[3] S. Russell & P. Norvig, Artificial Intelligence: A Modern Approach, 2021.

[4] The Holy Bible, New International Version, Zondervan, 2011, Exodus 20: 1-17.

[5] The Holy Bible, New International Version, Zondervan, 2011, Matthew 5:10.

[6] The Holy Bible, New International Version, Zondervan, 2011, Matthew 5:3-10.

[7] Future of Life Institute, Asilomar AI Principles, 2017.

[8] V. Dignum, The AI Paradox: How to Make Sense of a Complex Future, 2026.

[9] W. D. Cullen, The Ladbroke Grove Rail Inquiry Part 1 Report, UK Health and Safety Executive, 2021.

[10] The Holy Bible, New International Version, Zondervan, 2011, Exodus 20:8-11.

[11] The Holy Bible, New International Version, Zondervan, 2011, Matthew 5:3.

[12] E. Brymer, & L. G. Oades, “Extreme Sports: A Positive Transformation in Courage and Humility.” Journal of Humanistic Psychology, 49(1), 2009, DOI:10.1177/0022167808326199.

[13] J. Angwin, J. Larson, S. Mattu, & L. Kirchner, Machine Bias, ProPublica, 2016.

[14] The Holy Bible, New International Version, Zondervan, 2011, Exodus 20:12.

[15] The Holy Bible, New International Version, Zondervan, 2011, Matthew 5:4.

[16] B. L. M. Smits, G. J. Pepping & F. J. Hettinga “Pacing and Decision Making in Sport and Exercise,” Sports Medicine, 44, 763–775, 2014.

[17] National Transportation Safety Board, Collision Between Vehicle Controlled by Developmental Automated Driving System and Pedestrian. NTSB/HAR-19/03, 2019.

[18] The Holy Bible, New International Version, Zondervan, 2011, Exodus 20:13.

[19] The Holy Bible, New International Version, Zondervan, 2011, Matthew 5:5.

[20] S. Skorski, & C. R. Abbiss, “The Manipulation of Pace within Endurance Sport.” Frontiers in Physiology, 2017, 8:102. DOI:10.3389/fphys.2017.00102

[21] House Committee on Transportation and Infrastructure, The Design, Development & Certification of the Boeing 737 MAX. U.S. House of Representatives Staff Report, 2020.

[22] The Holy Bible, New International Version, Zondervan, 2011, Exodus 20:14.

[23] The Holy Bible, New International Version, Zondervan, 2011, Matthew 5:6.

[24] R. Hyland-Monks, L. Cronin, L. McNaughton, D. Marchant, “The role of executive function in the self-regulation of endurance performance.” Progress in Brain Research, 240, 353–370, 2018.

[25] Information Commissioner’s Office, Undertaking to the Information Commissioner: Royal Free NHS Foundation Trust, 2017.

[26] The Holy Bible, New International Version, Zondervan, 2011, Exodus 20:15.

[27] The Holy Bible, New International Version, Zondervan, 2011, Matthew 5:7.

[28] C. Thiel, K. Pfeifer & G. Sudeck, “Pacing and perceived exertion in endurance performance.” German Journal of Exercise and Sport Research, 48, 136–144, 2018.

[29] D. E. Hoffman, The Dead Hand: The Untold Story of the Cold War Arms Race and its Dangerous Legacy, New York: Doubleday, 2009.

[30] The Holy Bible, New International Version, Zondervan, 2011, Exodus 20:16.

[31] The Holy Bible, New International Version, Zondervan, 2011, Matthew 5:8.

[32] N. E. Brick, T. E. MacIntyre & M. J. Campbell, “Thinking and Action: A Cognitive Perspective on Self-Regulation during Endurance Performance.” Frontiers in Physiology, 7:159, 2016, DOI:10.3389/fphys.2016.00159.

[33] Associated Press, “Parents file lawsuit alleging AI chatbot contributed to teen harm,” 2024.

[34] The Holy Bible, New International Version, Zondervan, 2011, Exodus 20:17.

[35] The Holy Bible, New International Version, Zondervan, 2011, Matthew 5:9.

[36] A. McCormick, C. Meijen, P. A. Anstiss & H. S. Massey, “Self-regulation in endurance sports: theory, research, and practice.” International Review of Sport and Exercise Psychology, 12(1), 1–30, 2018.

[37] National Aeronautics and Space Administration, Resource Management on the Flightdeck: Proceedings of a NASA/Industry Workshop (NASA CP-2120), 1979.

[38] The Holy Bible, New International Version, Zondervan, 2011, Matthew 5.

[39] R. G. Cowden & E. L. Worthington, Overcoming Failure in Sport: A Self-forgiveness Framework. Journal of Human Sport and Exercise, 14(2), 2019, DOI:10.14198/jhse.2019.142.01. A. McCormick, C. Meijen, P. A. Anstiss & H. S. Massey, “Self-regulation in endurance sports: theory, research, and practice.” International Review of Sport and Exercise Psychology, 12(1), 1–30, 2018.

Harnessing Trustworthiness in LLM Agents through Embedding Trustworthy Engineering Life-Cycles into System Prompts

Sabrina Chaouche
IRT SystemX,
Palaiseau, France
email: sabrina.chaouche@irt-systemx.fr

Lucas Mattioli
IRT SystemX, Onera
Palaiseau, France
email: lucas.mattioli@irt-systemx.fr

Frédéric Barozzi
Naval Group
Toulon, France
email: frederic.barozzi@naval-group.com

Raphael Braud
IRT SystemX,
Palaiseau, France
email: raphael.braud@irt-systemx.fr

Faouzi Adjed
IRT SystemX,
Palaiseau, France
email: faouzi.adjed@irt-systemx.fr

Martin Gonzalez
IRT SystemX,
Palaiseau, France
email: martin.gonzalez@irt-systemx.fr

***Abstract*—Trustworthiness evaluation of Large Language Model (LLM)-based agents is currently predominantly metric-driven and use-case dependent. In most approaches, practitioners first define a task and subsequently select trust-related metrics, such as robustness, explainability, and statistical validity. We argue that this approach lacks grounding in established engineering life-cycles and quality processes. We propose a methodological inversion: instead of asking how to make an agent trustworthy, we begin with a well-defined trustworthy engineering process and embed this life-cycle directly into the system prompt of the agent. By structuring prompts around explicit stages, actors, and Responsible, Accountable, Consulted, Informed (RACI) matrices, trust becomes a matter of process compliance rather than post hoc output evaluation. We illustrate our approach with a ReAct-style (Reasoning and Acting) data analyst agent and show how stage-specific automation enables principled trade-offs between full automation and human oversight. This reframing positions agents as instruments of controlled and trusted processes rather than autonomous endpoints.**

Keywords-Trustworthy AI; LLM Agents; Engineering Life-Cycle; System Prompt; Data Governance.

I. Introduction

Large Language Models (LLMs) are increasingly deployed not only as conversational systems but as agents capable of reasoning, tool use, and multi-step task execution. Architectures combining language reasoning and external actions, such as ReAct [1] have demonstrated that LLMs can orchestrate data processing pipelines, call external tools, write code, and iteratively refine their outputs, accelerating their integration into domains traditionally governed by structured engineering processes.

As LLM-based agents move into such operational contexts, the question of trustworthiness becomes central. Major governance frameworks—such as those of the European Commission and the Organisation for Economic Co-operation and Development (OECD) have articulated high-level requirements including robustness, accountability, transparency, and human oversight [2]. In practice, trustworthiness of LLM-agent is typically assessed by decomposing a task into subcomponents and selecting appropriate metrics for each subtask.

While valuable, the metric-centric, commonly used, approach exhibits a structural limitation: trustworthiness is treated as a property of outputs, evaluated *after* task execution. This implicitly assumes that trust can be derived from aggregating measurable output properties. However, in many professional domains, trust is not grounded solely in output correctness, but in **procedural compliance** with established engineering life-cycles. In data analysis, software engineering, or quality management, an output is considered trustworthy not merely because it appears correct, but because it has been produced *according to a recognized, auditable process* defined independently of any specific automation technology.

This paper proposes a methodological inversion. Instead of asking: *Given an agent, which trust metrics should we use to evaluate it for a given task? How do we train a trustworthy agent?* We ask: *Given a principled trustworthy engineering process, how can an LLM-based agent be embedded within this process so as to automate specific stages while measuring the amount to which it preserves procedural guarantees?*

We shift the focus from *agent-centered trust calibration* to **process-centered trust embedding**. LLM-based agents are treated not as autonomous systems whose trustworthiness must be independently established, but as instruments operating within a pre-defined quality process. Trust is derived from the degree to which the agent complies with, documents, and enables monitoring of the stages of a recognized engineering life-cycle.

The central technical mechanism enabling this inversion is the integration of the engineering life-cycle into the *system prompt* of the LLM-based agent. We argue that the system prompt should not merely define the agent's role (e.g., "You are a data analyst"), but should explicitly encode: (i) the stages of the relevant life-cycle; (ii) the objectives and expected intermediate outputs of each stage; (iii) the allocation of responsibilities across actors; and (iv) the conditions under which automation should stop and human intervention should occur. By embedding this information directly into the agent's operational context, we transform the system prompt into a governance artifact. The agent's outputs can then be evaluated

 ISBN: 978-1-68558-395-8

relative to clearly defined stage-specific criteria, rather than abstract task-level expectations.

We illustrate this approach through a case study involving a simple LLM-based data analyst agent. Data analysis is a particularly suitable domain for this exploration because it is governed by well-established methodological standards: statistical inference requires explicit assumptions and significance reporting; regression analysis requires model specification and validation; and communication of results requires appropriate visualization and documentation.

The contributions of this paper are threefold:

1) We provide a conceptual reframing of trustworthiness in LLM-based agents as compliance with established engineering life-cycles rather than as a collection of post hoc metrics.
2) We formalize a process-centric framework in which life-cycle stages, actors, and validation criteria are embedded into enriched system prompt templates.
3) We demonstrate the approach in the concrete setting of a data analysis agent, highlighting how stage-dependent automation enables principled trust trade-offs.

By reversing the prevailing perspective—from "training or evaluating trustworthy agents" to "embedding agents within trustworthy processes"—we aim to reposition LLM-based agents as means within structured governance frameworks, rather than as ends in themselves.

The rest of the paper is structured as follows. In Section II, we review related work on trustworthy LLM-agent evaluation frameworks and motivate the need for process-centric methodologies. In Section III, we introduce our methodological inversion, formalizing the trustworthy engineering life-cycle and the role of the system prompt as a process-encoding mechanism. In Section IV, we instantiate the framework through a trustworthy data analytics life-cycle comprising ten stages, and we define actor roles via a RACI matrix. In Section V, we describe stage-relative trustworthiness attributes and associated metrics. In Section VI, we illustrate our approach through a ReAct-style data analyst agent and discuss stage-level compliance auditing. Finally, Section VII concludes the paper and outlines directions for future work.

II. Related Work

LLM-based agents have the capabilities of autonomy and reactivity in their decisions, as well as interactivity and usability with other tools during reasoning and planning phases [3]. However, the assessment of the trustworthiness of these agents becomes more complex and challenging [4]. In the literature, several trustworthiness frameworks for LLMs-based agents evaluations are proposed. A framework analysis by Yu et al. [4] considers modular taxonomy, multi-dimensional connotation, and technical implementation. MLA-Trust framework, proposed by Yang et al. [5], studies truthfulness, controllability, safety, and privacy. TrustAgent framework, proposed by Hua et al. [6], focuses on the safety assessment by injecting safety knowledge and planning strategy into the evaluation framework. More recently, Bamil et al. [7] introduced a unified evaluation and governance framework that integrates decision controls directly within the LLM agent inference loop.

However, traditional engineering practices, such as structured life-cycles and responsibility frameworks, are rarely integrated into the design of LLM-based agent assessments. This highlights the need for methodologies that explicitly define structured life-cycles to ensure reliability, traceability, and accountability. Such life-cycles are based on workflows decomposed into well-defined stages, each associated with explicit objectives, required actions, validation criteria, and assigned responsibilities. A key governance instrument within these processes is the Responsible, Accountable, Consulted, and Informed (RACI) matrix [8], which formalizes role allocation across stakeholders. For instance, in a data analytics workflow, a data analyst may be responsible for descriptive analysis, while a domain expert remains accountable.

The key contribution in the current work is to enable an agent to work inside a controlled and trusted process that avoid any random action from it. Therefore, an agent needs to follow a trustworthy engineering stages by introducing a clear prompt structure, executing all the important steps, and producing intermediate results in addition to the final result. In the current approach, we propose to include the whole prompting process that an agent must follow.

III. Methodological Inversion: Trust as Life-Cycle Compliance

A. From Metric-Centric Evaluation to Process-Centric Design

The prevailing approach to trustworthiness in LLM-based agents is metric-centric: designers define a task, select trust metrics (e.g., accuracy, robustness), execute the agent, and evaluate the output *post hoc*. This paradigm is reflected in high-level governance frameworks such as those of the European Commission and the OECD, which articulate requirements and principles but do not prescribe how these are operationally embedded into the runtime structure of agent systems.

In this classical setting, trustworthiness is treated as a property of outputs. The methodological order is: 1) define a task or use-case, 2) select relevant trust metrics, 3) execute the agent, and 4) evaluate outputs against the chosen metrics.

This approach has two limitations. First, the selection of metrics is often contextual but insufficiently grounded in a structured engineering methodology. Second, evaluation occurs post hoc: trust is measured after the agent has acted, rather than being structurally integrated into the conditions of its action.

We propose a methodological inversion. Instead of deriving trust requirements from tasks and evaluating outputs accordingly, we begin from a formally specified trustworthy engineering life-cycle and situate the agent within it. Trustworthiness is thus redefined as compliance with a pre-specified process, rather than as an aggregate property of isolated outputs.

B. Trustworthy Engineering Life-Cycle as Primary Object

Let L denote a trustworthy engineering life-cycle composed of ordered stages:

$$L = \{S_1, \ldots, S_n\}, \quad \text{with } S_i = (O_i, A_i, V_i, R_i)$$

being a stage consisting of an objective O_i, specified required actions A_i, validation criteria or success conditions V_i, and role allocation R_i (e.g., via a RACI matrix).

This formalization reflects established quality engineering practices in which processes are decomposed into stages with explicit responsibilities and validation checkpoints. Crucially, these life-cycles are independent of any specific automation technology. They are normative descriptions of how a trustworthy outcome ought to be produced. Under our approach, the life-cycle L becomes the primary design object. The LLM-based agent Agent is introduced only as a potential executor of one or more stages within L.

C. System Prompt as Process-Encoding Mechanism

The central operational mechanism of our approach is the integration of the life-cycle specification L into the system prompt of the agent. Indeed, conventionally, system prompts define (i) role, (ii) behavioral constraints, and (iii) tool usage instructions. In our approach, the system prompt additionally encodes (i) ordered stages $S_1, \ldots, S_n$, (ii) the objective O_i of each stage, (iii) expected intermediate outputs, (iv) validation requirements V_i, (v) role and responsibility constraints R_i, and (vi) explicit stopping or escalation conditions.

The system prompt thus becomes a *methodological container* that situates the agent within a structured engineering process. It does not merely instruct the agent *what* to do, but *how the action fits into a larger trust-governed procedure*.

Formally, let $P(L)$ denote the prompt encoding of life-cycle L. The behavior of the agent becomes a function of both user input U and process specification:

$$\mathrm{Output} = \mathrm{Agent}(U, P(L)).$$

Without $P(L)$, the agent operates without explicit awareness of the normative process constraints governing its outputs. We therefore argue that: *Providing the life-cycle context in the system prompt is a necessary condition for evaluating compliance-based trustworthiness in LLM-based agents.*

D. Monitoring Through Intermediate Outputs

A further methodological consequence of embedding L into the system prompt is the systematic production of intermediate outputs.

For each stage S_i, the agent is required to generate:

- A structured report of actions taken,
- Justification of methodological choices,
- Explicit uncertainty statements,
- Artifacts enabling external validation (e.g., code, statistics, structured data).

These intermediate outputs serve two purposes: first they enable monitoring of compliance with V_i, secondly, they allow selective interruption of automation.

Trust evaluation thus shifts from inspecting final answers to auditing stage-level artifacts. This enables principled trade-offs. If compliance at stage S_i is systematically insufficient (e.g., low-quality visualizations), designers may reassign that stage to human execution, without discarding automation at other stages.

E. Automation as Stage-Selective Delegation

Let $\mathcal{A} \subseteq L$ denote the subset of stages delegated to the LLM-based agent. For each stage S_i, we define an automation coefficient: $\alpha(S_i) \in [0, 1]$, where $\alpha(S_i) = 1$ signifies full automation, $\alpha(S_i) = 0$ indicates exclusive human execution, and $0 < \alpha(S_i) < 1$ represents a hybrid or semi-automated approach. Although the determination of α is intrinsically context-dependent, for the purposes of analytical clarity within this paper, we adopt a baseline of $\alpha = 0.5$. Future research will address more sophisticated formalizations of these automation scores in greater detail.

This framework underscores that automation is not global but stage-specific. Trustworthiness therefore cannot be assessed solely at the system level; it must be evaluated relative to each stage's objectives and validation criteria.

An agent is trustworthy with respect to stage S_i if and only if:

1) It produces outputs consistent with O_i,
2) It performs actions consistent with A_i,
3) Its outputs satisfy validation constraints V_i,
4) It respects the role allocation constraints in R_i.

Trust thus becomes a relation:

$$\mathrm{Trust}(\mathrm{Agent}, S_i) \iff \mathrm{Compliance}(\mathrm{Agent}, S_i).$$

Global trustworthiness is then a function over all delegated stages:

$$\mathrm{Trust}(\mathrm{Agent}, \mathcal{A}) = f\left(\{\mathrm{Compliance}(\mathrm{Agent}, S_i) \mid S_i \in \mathcal{A}\}\right).$$

This formulation replaces metric aggregation with structured process compliance.

F. Re-framing the Agent's Epistemic Status

Under the metric-centric paradigm, the agent is the primary epistemic object: the system is evaluated as trustworthy or untrustworthy according to the following approach:

Classical approach
Task $\rightarrow$ Agent $\rightarrow$ Output $\rightarrow$ Metrics $\rightarrow$ Trust assessment

Under our process-centric paradigm, the engineering life-cycle is primary. The agent is a tool embedded within a normative structure:

Proposed approach
Trustworthy life-cycle $\rightarrow$ Stage allocation $\rightarrow$ Prompt encoding $\rightarrow$ Stage-level compliance $\rightarrow$ Trust assessment

Tool-based Methodology. One can argue that if stage-level compliance is provided by prompting the agent to export its own history of actions, that it might very well hallucinate or even hide the actual actions it effectively made. For this reason, we introduce a further numerical artifact called **Action Tracker** that consists of an external layer that logs all calls, observations and changes in the planning that the agent makes, providing an *independent* action's history that will further capture the LLM-based agent's attempts to hallucinate or hide actions (see Figure 1).

By embedding established engineering methodologies directly into system prompts, we transform trust from an *ex-post* evaluation criterion into an *ex-ante* structural constraint. As such, Trust is not a static attribute of the agent; it is a dynamic property of the interaction between:

- The life-cycle L,
- The allocation function α,
- The compliance behavior of the agent.

This reframing has two implications:

1) Trustworthiness becomes contextual *and* stage-relative.
2) The design question shifts from "How do we build a trustworthy agent?" to "Given a trustworthy process, how do we allocate stages to an agent while preserving compliance?"

IV. Case Study: Trustworthy Data Analytics

In this section, we focus on the particular case of a trustworthy data analysis life-cycle, which has well documented methodological approaches. First, we recall the latter as a reference and we enounce their strengths and limitations. Second, we propose a synthesis of such methodological approaches for data analytics in 10 steps, leveraging their strengths with regards to trustworthiness, we determine different actors and a RACI matrix for process allocation and responsibility level. This will be the basis upon we frame our LLM-based agent.

A. Related Methodological Approaches

The Data Analytics Life Cycle (DALC) provides a structured framework for navigating the journey from raw data to actionable insights. Modern methodologies increasingly prioritize **trustworthiness**—a multi-dimensional concept encompassing ethics, reliability, and governance.

Classic & Industry Standard Models

- **Knowledge Discovery in Databases (KDD):** The foundational academic framework [9]. It focuses on the technical evolution: Selection $\rightarrow$ Preprocessing $\rightarrow$ Transformation $\rightarrow$ Data Mining $\rightarrow$ Interpretation/Evaluation.
- **Cross-Industry Standard Process for Data Mining (CRISP-DM):** The most widely used industry framework [10]. It uses a cyclical approach with six phases: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment.
- **Sample, Explore, Modify, Model, Assess (SEMMA):** Developed by SAS Institute (Sample, Explore, Modify, Model, Assess), focusing heavily on the technical modeling cycle [11].

Modern & Governance-Focused Models

- **Veridical Data Science (PCS Framework):** Proposed by Yu et al. [12], centering on **Predictability, Computability, and Stability**. It provides a rigorous mathematical basis for ensuring results are reproducible.
- **National Institute of Standards and Technology (NIST) AI Risk Management Framework (AI RMF):** A 2023 framework designed to manage risks and promote Trustworthy AI [13]. It focuses on four core functions: Govern, Map, Measure, and Manage.
- **Cross-Industry Standard Process for Machine Learning with Quality Assurance (CRISP-ML(Q)):** A recent extension [14] of CRISP-DM that explicitly incorporates quality assurance and fairness audits into each traditional phase.

B. Our 10-Stage Trustworthy Data Analytics Life Cycle

We provide an expanded life-cycle approach for Trustworthy Data Analytics that develops on top of the foundational structure of CRISP-DM, the technical rigor of the PCS framework (Yu & Barter, 2024), and the risk-centrality of the NIST AI Risk Management Framework.

TABLE I. RACI Matrix for the 10-Stage Life Cycle (R: Responsible, A: Accountable, C: Consulted, I: Informed).

Stage	BS	DE	DS	DA	ECO	MLO
1. Governance & Scope	**A**	I	C	C	**R**	I
2. Discovery & Acquisition	I	**R/A**	C	C	C	I
3. Data Validation	I	**R/A**	I	**R**	I	C
4. (EDA1) Descriptive	C	I	C	**R/A**	I	I
5. (EDA2) Diagnostic	C	I	**R**	**R/A**	I	I
6. (EDA3) Predictive	I	I	**R/A**	C	I	C
7. (EDA4) Prescriptive	**A**	I	**R**	C	C	I
8. Trustworthiness Audit	C	I	C	I	**R/A**	C
9. Communication & Viz	**R**	I	I	**R/A**	C	I
10. Doc & Reproducibility	I	**R**	**R**	I	I	**A**

1. Governance and Scope: Defining the problem statement, business objectives, and success metrics while establishing the ethical and legal boundaries for the project.

2. Data Discovery and Acquisition: Identifying internal/external data assets, negotiating access, and documenting data lineage and provenance.

3. Data Validation: Rigorous checking of data integrity, schema consistency, and quality constraints to ensure the "raw" material is fit for purpose.

4. (EDA1) Descriptive Analytics: Summarizing the historical "ground truth" through statistical profiling and trend analysis to report what has occurred.

5. (EDA2) Diagnostic Analytics: Investigating causal factors and root causes to explain the "why" behind the patterns observed in EDA1.

6. (EDA3) Predictive Analytics: Developing and validating mathematical models to forecast future outcomes based on historical and diagnostic features.

7. (EDA4) Prescriptive Optimization: Using optimization techniques and decision logic to recommend the best course of action based on predictions.

8. Trustworthiness Audit: A systematic review for bias (Fairness), sensitivity to perturbations (Stability), and vulnerability to adversarial threats (Robustness).

9. Communication and Visualization: Translating complex analytical outputs into intuitive, stakeholder-aligned visual narratives and actionable insights.

10. Documentation and Reproducibility: Finalizing the "analytical paper trail" (code, environment, and metadata) to ensure any third party can replicate the results.

C. Actor Definitions and RACI Matrix

The life cycle is supported by six primary human roles:

- **BS (Business Stakeholder):** Owns the business problem and the ultimate value realization.
- **DE (Data Engineer):** Manages the flow, storage, and architectural integrity of data.
- **DS (Data Scientist):** Builds the predictive and prescriptive logic; focuses on modeling.
- **DA (Data Analyst):** Focuses on discovery, descriptive reporting, and diagnostic insights.
- **ECO (Ethics & Compliance Officer):** Ensures adherence to legal, regulatory, and moral standards.
- **MLO (MLOps / IT Engineer):** Manages the technical infrastructure, deployment, and reproducibility.

Comparison of Trustworthiness Attributes

Table II evaluates how explicitly each approach addresses core pillars of trustworthiness.

- **Explicit:** Primary goal with defined tasks/metrics.
- **Limited:** Mentioned or implied without formal procedures.
- **Absent:** Not addressed in original documentation.

V. Stage-Relative Trustworthiness Attributes and Metrics

Within the proposed process-centric framework, trustworthiness is not evaluated globally but relative to a specific stage S_i of the engineering life-cycle. For a given stage S_i, we define a set of trustworthiness attributes:

$$\mathcal{T}(S_i) = \{T_{i1}, T_{i2}, \ldots, T_{ik}\},$$

each associated with measurable metrics M_{ij} evaluated conditionally on the objective and validation requirements of S_i. We identify eight core attributes:

Methodological Compliance (all stages) captures the degree to which the agent follows the prescribed methodological steps of stage S_i. Typically measured through stage declaration accuracy (binary indicator), the required artifact coverage ratio, a validation rule satisfaction rate, or an out-of-scope action count. This attribute is central, as compliance with the life-cycle is a necessary condition for trust under our framework.

Statistical Validity (S4, S5) measures adherence to accepted inferential standards through test completeness scores (presence of hypothesis, statistic, p-value, effect size, etc.), assumption reporting indicator, and uncertainty quantification presence indicator. These metrics are required in diagnostic stages but not necessarily in earlier ones.

Reproducibility (S5, S7) assesses whether outputs allow independent replication, tracked via executable code availability (binary indicator), parameter transparency proportion score, and artifact sufficiency index.

Interpretability and Justification (exploratory, inference, and modeling stages) evaluates the degree to which analytical choices are justified and interpretable, through justification presence binary indicators, coefficient interpretation completeness, and the explicit presence of the correlation-causation distinction. This attribute ensures that outputs remain epistemically disciplined within the stage objective.

Uncertainty Transparency (inference and modeling stages) measures the degree to which limitations are reported, using limitation disclosure indicators, uncertainty coverage ratio, and assumption explicitness scores.

Data Integrity and Quality Awareness (S2, S3) captures the degree to which the agent identifies data quality constraints, including missing values, sample sizes, potential biases, and outlier impact assessments. These ensure downstream stages are not executed on unexamined data foundations.

Visualization Integrity (S6) assesses whether visual artifacts are accurate, interpretable and non-misleading, verified through axis label completeness (binary indicator), a legend presence indicator, scale appropriateness, and a plot-data consistency score.

Role Compliance (all stages) verify if the agent respects the role allocation for a given stage S_i by checking escalation compliance, respect of human-required boundaries, and appropriate consultation triggers. This attribute makes governance operational within the life-cycle

Stage-Level Trust Score For a given stage S_i, a composite trust score is defined as $\mathrm{Trust}(\mathrm{Agent}, S_i) = g(T_{i1}, \ldots, T_{ik})$, where g is a domain-specific aggregation function. Failure in methodological compliance may invalidate a stage regardless of statistical correctness.

These attributes differ from classical benchmarks in three ways: they are stage-relative (and not global), they evaluate process compliance rather than output correctness alone, and they enable automation boundary tuning through selective monitoring.

VI. Illustration and discussion: ReAct Data Analyst Agent

This section demonstrates the proposed methodology through a concrete use case: the automation of specific stages within a trustworthy data analysis lifecycle. The implementation employs a ReAct-style, LLM-based agent configured to

TABLE II. COMPARATIVE ANALYSIS OF TRUSTWORTHINESS PILLARS ACROSS DALC MODELS.

Methodology	Transparency	Accountability	Fairness	Robustness	Privacy
KDD	Limited	Absent	Absent	Limited	Absent
CRISP-DM	Limited	Limited	Absent	Limited	Absent
SEMMA	Absent	Absent	Absent	Explicit	Absent
TDSP	Explicit	Explicit	Limited	Explicit	Limited
Veridical (PCS)	Explicit	Limited	Limited	Explicit	Absent
NIST AI RMF	Explicit	Explicit	Explicit	Explicit	Explicit
CRISP-ML(Q)	Explicit	Explicit	Limited	Explicit	Limited
TDAP (ours)	Explicit	Explicit	Explicit	Explicit	Explicit

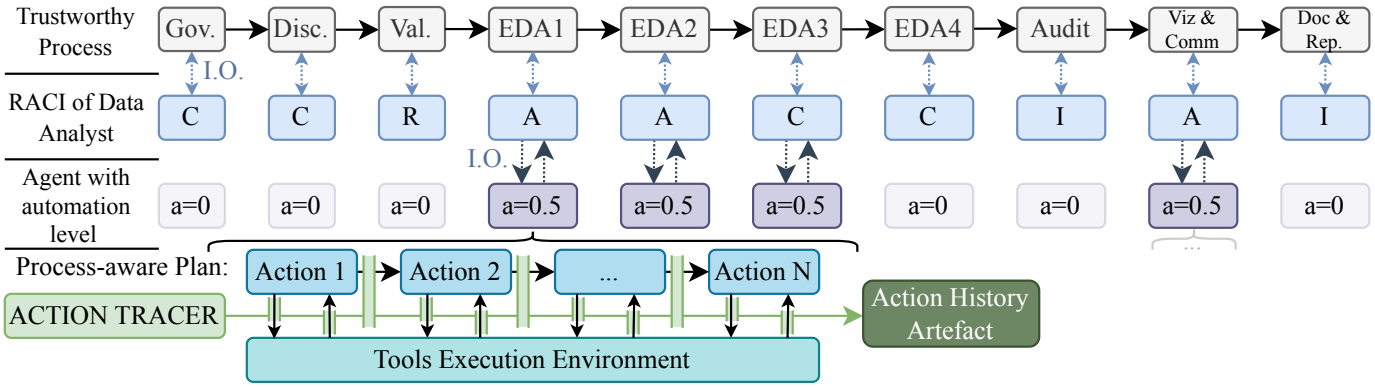


Figure 1. Recapitulation scheme of our framework: An actor plays a specific stage-wise role in a trustworthy process, and is assisted by an agent at a determined automation level pertinent to each stage. For stages with non-zero automation level, the agent is then doubly constrained top-down by the process (i.e. system prompt) & bottom-up by the action tracker artifact that guarantees the independent tracking of actions, observations, re-planing the agent does.

function as a *data analyst*. In this scenario, the agent is tasked with analyzing US Census data via the Folktables dataset [15], a benchmark framework derived from the American Community Survey (ACS). Rather than evaluating performance metrics in isolation, this illustration aims to show how embedding a formal lifecycle into the system prompt governs both execution and evaluation. Consequently, the agent is situated within a formally specified engineering process to assess its stage-level compliance.

A. Baseline: Task-Centric Evaluation of a ReAct Agent

In a metric-centric setting, a ReAct-style agent would be tasked with discrete analytical operations on socio-economic records from the *Folktables* dataset, which contains features, such as income level, employment status, and demographic covariates. Typical prompts in such a scenario might include:

1) **Descriptive Analysis:** "Calculate the median and mean annual income for each level of educational attainment for the year 2018. Present the results in a summary table sorted by education level, and identify which group exhibits the highest income variance."
2) **Statistical Comparison:** "Investigate the gender pay gap among full-time workers. Report the t-statistic, the p-value, and provide a box-plot visualizing the income distributions for both groups."
3) **Predictive Modeling:** "Construct a regression model to predict an individual's annual income. Use age, hours worked, and gender as the primary predictors. After fitting the model using *scikit-learn*, report R^2 score and the coefficients for each feature. Finally, provide a residual plot to assess the model's heteroskedasticity."

While the evaluation would focus on:

- The correctness of computed statistics,
- The validity of statistical test selection,
- The executability and syntactic correctness of the generated code,
- The interpretability and visual clarity of the resulting plots.

While these criteria are necessary, they remain fragmented and detached from a structured engineering process. Consequently, the agent is judged primarily by the adequacy of its final output rather than its adherence to a compliant and trustworthy analytical procedure.

B. Task-Centric Evaluation to Process-Centric Allocation

In our approach, the same ReAct agent is configured with an enriched system prompt encoding a formally specified trustworthy data analysis life-cycle. The agent is instructed to:

(i) Explicitly identify the stage of the life-cycle being executed, (ii) Produce required intermediate artifacts, (iii) Validate assumptions before proceeding, (iv) Signal uncertainty and limitations, (v) Respect automation boundaries defined per stage.

The tasks are no longer treated as isolated queries. Instead, they correspond to specific stages of the life-cycle, the agent is then delegated a role (in the RACI sense) for each, and its outputs are evaluated for compliance with stage objectives and validation criteria.

- Income distribution across educational levels → Descriptive Analytics (S4),
- Hypothesis testing for gender-based income disparities → Statistical Inference (S5),
- Multivariate-regression of socio-economic income determinants → Modeling (S6),
- Boxplot and residual plot → Visualization (S9).

The subsequent stages remain under human accountability and oversight.

C. Process Allocation and Responsibility Model

The purpose of this section is twofold: (i) demonstrate how life-cycle embedding structures the agent's behavior, and (ii) show how trustworthiness is evaluated as stage-level compliance rather than global output correctness.

Let $L = \{S_1, \ldots, S_{10}\}$ denote the life-cycle above. We define the non-zero automation subset:

$$\mathcal{A} = \{S_4, S_5, S_6, S_9\}.$$

For each $S_i \in \mathcal{A}$, the agent is either *Responsible* or *Consulted*, while a human expert remains *Accountable*. This allocation specification ensures that:

- Methodological decisions remain reviewable.
- Intermediate artifacts are auditable (externally, leveraging the *action tracker* artifact).
- Automation boundaries can be flagged to be readjusted stage-wise.

In the following, we specify the stages within the agent's operational scope.

1) Stage 4 (EDA1) - Descriptive Analytics:

a) Objective: Provide a structured quantitative characterization of the data.

b) Required Outputs: When computing the median and mean annual income per level of educational attainment for the year 2018, the agent must:

- Report sample size per educational level,
- Provide structured tabular output,
- Clearly distinguish descriptive statistics from inferential claims and causal interpretation.

c) Trust Constraints:

- No inferential claims at this stage.
- Clear distinction between numerical reporting and interpretation.
- Explicit identification of anomalies or irregularities.

Compliance is verified by checking that all descriptive indicators are numerically reported and that no unwarranted causal or inferential statements are introduced.

In Figure 1, we show exactly how the relationship between the LLM-Agent and our framework is done for *Stage 4*: we begin by identifying the human actor's tasks as framed within a clear process, we proceed by identifying the standards and norms for such process to be compliant with quality requirements allowing us to characterize it as being trustworthy-as-a-process; we spread the process' life-cycle stages, enumerate the involved stakeholders in the process with their RACI roles (one of which is our initial actor); for a given stage where the agent's automation level is non-zero, we provide the agent all the above information in the form of a system prompt and let the agent make an initial plan contextualized as being relative to all preceding specifications; we activate the action tracker artifact that will log all *effective* actions, calls, observations, plan's redesign that the agent makes while providing assistance to the actor to the extent of the specified objective, required actions, validated criteria associated to the stage and which are finally kept as a history artifact that will later allow external audit of the LLM-Agent.

2) Stage 5 (EDA2) - Diagnostic Analytics:

a) Objective: Explain observed patterns through statistical comparison and hypothesis testing.

b) Required Outputs: When investigating gender-gap income disparities, compliance requires:

- Explicit null and alternative hypotheses.
- Justification of test selection.
- Reporting of test statistics and p-values.
- Reporting of samples sizes.
- Assumption checks (e.g., distributional assumptions, independence, normality).

c) Trust Constraints:

1) No p-value without reporting the corresponding statistic test.
2) No statistical significance claim without effect size.
3) Assumptions must be explicitly stated.
4) Correlation must not be framed as causation.

In this stage, trustworthiness hinges on methodological correctness and epistemic transparency: a statement such as "the difference is statistically significant" without test statistics constitutes a methodological failure. The agent's compliance is evaluated not only by correctness of results but by adherence to statistical reporting standards.

3) Stage 6 (EDA3) - Predictive Analytics:

a) Objective: Construct and interpret predictive models (e.g., regression).

b) Required Outputs:

- Explicitly define the predictive features,
- Justify the choice of predictors,
- Provide reproducible Python code,
- Report numerical coefficients,
- Interpret coefficients cautiously,
- Provide a clear diagnostic commentary, distinguishing correlation from causation.

c) Trust Constraints:

1) Clear separation between prediction and explanation.
2) No causal interpretation without identification strategy.

3) Full reporting of coefficients before interpretation.
4) Explicit acknowledgment of uncertainty.

Compliance is assessed not only by whether the code runs, but by whether the modeling stage adheres to accepted analytical standards.

4) Stage 9 - Communication and Visualization:

a) Objective: Communicate analytical findings in a manner consistent with transparency and non-misleading representation.

b) Trust Constraints:

- Axes must be labeled.
- Units must be specified.
- Scaling must not distort interpretation.
- Uncertainty must be visually or numerically represented.

If visual compliance cannot be guaranteed, the actor must flag and reduce automation, and ask the LLM-agent to provide intermediate artifacts instead in order for the human actor to have everything ready to guarantee visual compliance. We do this by explicitly prompting defines acceptable intermediate artifacts and escalation conditions. As such this constitutes a genuine **trade-off** between automation and compliance, in contrast to the situation where efficiency, as provided by unsupervised full automation, and compliance, is not a valid trade-off as it breaks the assumption that LLM-agents cannot hold an accountability position on the process.

D. Output Evaluation to Stage-Level Compliance Auditing

The case study demonstrates that trust assessment shifts from global performance scoring to stage-level compliance auditing. The ReAct data analyst agent is repositioned from an autonomous answer generator to an instrument embedded within a structured, monitorable engineering process: the same numerical output may be deemed compliant or non-compliant depending on whether the agent respected the validation rules of the corresponding stage. Concretely, evaluation criteria include whether the agent identifies and respects the current stage, produces the required intermediate artifacts, satisfies stage-specific validation conditions, and transparently reports uncertainty and limitations.

This operationalization yields two central properties of the proposed framework: trustworthiness is **stage-relative** rather than global; and compliance is **auditable** through structured intermediate artifacts. Rather than asking whether the agent is trustworthy in the abstract, evaluation assesses whether it complies with the procedural constraints of each life-cycle stage; EDA1, EDA2, EDA3, and Communication and Visualization. This operationalizes the methodological inversion introduced earlier: trustworthiness is not inferred from final outputs alone, but derived from a governance-aware analytical process.

VII. Conclusion and Future Work

In this work, we propose a methodological inversion in evaluating LLM-based agent trustworthiness: rather than assessing aggregate outputs, agents are embedded as instruments within governance-aware engineering processes. Under this reframing, trustworthiness becomes a measurable function of procedural compliance, and the system prompt is elevated from a task descriptor to a governance artifact. Evaluation consequently shifts from final outputs to stage-level compliance auditing, enabling human oversight to be calibrated per stage; though effective instantiation requires substantial domain expertise that cannot itself be delegated to the agent. Several limitations warrant acknowledgment. The framework formalizes a sequential life-cycle without explicit feedback loops. Additionally, embedding full life-cycle specifications into system prompts introduces non-trivial complexity. Future work should investigate iterative and multi-agent extensions, stage-level compliance verification for stages with non-zero automation allocation and systematic evaluation across diverse engineering contexts.

Acknowledgment

This work has been supported by the French government under the "France 2030" program, as part of the SystemX Technological Research Institute within the CSIA Project. We thank Emna Amdouni for useful & fruitful discussions.

References

[1] S. Yao et al., "React: Synergizing reasoning and acting in language models", in *The eleventh international conference on learning representations (ICLR)*, 2023.

[2] M. Adedjouma et al., *Towards the Engineering of Trustworthy AI Applications for Critical Systems. The Confiance.ai Program*, pp. 9–12, 2022.

[3] Z. Xi et al., "The rise and potential of large language model based agents: A survey", *Science China Information Sciences*, vol. 68, no. 2, pp. 1–38, 2025.

[4] M. Yu et al., "A survey on trustworthy llm agents: Threats and countermeasures", in *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, 2025, pp. 6216–6226.

[5] X. Yang et al., "Mla-trust: Benchmarking trustworthiness of multimodal llm agents in gui environments", *arXiv preprint arXiv:2506.01616*, 2025.

[6] W. Hua et al., "Trustagent: Towards safe and trustworthy llm-based agents", in *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024, pp. 10 000–10 016.

[7] V. Bamil et al., "A unified evaluation and governance framework for trustworthy llm agents", *Authorea Preprints*, 2026.

[8] T. Pitkäranta and L. Pitkäranta, "Hada: Human-ai agent decision alignment architecture", in *International Joint Conference on Computational Intelligence*, Springer, 2025, pp. 78–102.

[9] U. Fayyad et al., "From data mining to knowledge discovery in databases", *AI magazine*, vol. 17, no. 3, 1996.

[10] P. Chapman et al., "Crisp-dm 1.0: Step-by-step data mining guide", SPSS Inc., Tech. Rep., 2000.

[11] A. Azevedo and M. F. Santos, "Kdd, semma and crisp-dm: A parallel overview", in *IADIS European Conference on Data Mining*, 2008, pp. 182–185.

[12] B. Yu et al., *Veridical Data Science: The Statistics, Prediction and Algorithms (PCS) Framework*. MIT Press, 2024.

[13] National Institute of Standards and Technology, "Artificial intelligence risk management framework (ai rmf 1.0)", U.S. Department of Commerce, Tech. Rep., 2023.

[14] S. Studer et al., "Towards crisp-ml (q): A machine learning process model with quality assurance methodology", *Machine Learning and Knowledge Extraction*, vol. 3, no. 2, 2021.

[15] F. Ding et al., "Retiring adult: New datasets for fair machine learning", *NeurIPS*, vol. 34, 2021.

Evaluating Performance, Safety, and Robustness of an AI-Based Airport Delay Alerting Tool with Calibrated Machine Learning for Operational Decision Support

Soufiane Momtaz
ENSET Mohammedia, Hassan II University
Casablanca, Morocco
Email: soufiane.momtaz@gmail.com

Otmane Idrissi
ENSET Mohammedia, Hassan II University
Casablanca, Morocco
Email: iodrimane@gmail.com

Joseph Machrouh
Thales LAS France
Rungis, France
Email: joseph.machrouh@thalesgroup.com

***Abstract*—Artificial intelligence (AI)-based decision-support tools in airport operations should be assessed not only by predictive performance but also by the safety and robustness of the alerting policies they induce under temporal drift. This paper presents a leakage-free one-month-ahead case study on United States (U.S.) Bureau of Transportation Statistics (BTS) airport-carrier-month data from 2022–2024 and makes four contributions. First, this work specifies a system pathway from pre-month data to calibrated probabilistic scoring, thresholded alerting, and deployment audit. Second, this work introduces an assurance-oriented evaluation protocol that combines the area under the receiver operating characteristic curve (AUC), Brier score, and log-loss for prediction; nominal-threshold decision risk for safety; and Threshold-Local Calibration Error (TLCE), Action-Overconfidence Gap (AOG), and risk stability across thresholds for robustness. Third, this work compares logistic regression, random forest, and extreme gradient boosting (XGBoost) under a strict train–calibration–test chronology and evaluates Platt, isotonic, and beta calibration under temporal transfer, positioning these tabular baselines against state-of-the-art delay-prediction practice. Fourth, this work discusses scalability and security-aware operational integration and shows through monthly audit that calibration behavior changes across deployment sub-regimes. The main conclusion is that calibration is a time-sensitive operational component that must be validated through local diagnostics, threshold-aware reporting, and continuous deployment auditing.**

Keywords-safety; robustness; performance; artificial-intelligence-based systems; temporal drift; calibration; airport delay alerting.

I. Introduction

Artificial intelligence (AI)-based systems are increasingly used to issue alerts, prioritize interventions, and support planning in airport and transportation operations. Once a probabilistic score is thresholded into an action, evaluation should cover not only ranking ability but also the operational consequences of acting or not acting on that score. In this paper, *performance* denotes predictive quality, *safety* denotes cost-weighted decision behavior under missed and unnecessary alerts, and *robustness* denotes stability across thresholds and changing operating conditions. The aim is to support assurance-oriented evaluation of AI-based decision-support tools for aviation operations, not to claim certification of the studied prototype.

Operational evaluation should separate probabilistic scores from the action rules and costs they induce. Threshold-insensitive summaries, such as the area under the receiver operating characteristic curve (AUC), can hide materially different operational behaviors [1]. Calibration and probabilistic-forecasting studies also show that strong discrimination does not guarantee reliable probabilities [2]–[9]. Under dataset shift, uncertainty quality can deteriorate, which is a major issue for the safe operational use of AI systems [10].

This paper revisits airport delay analysis from an operational decision-making standpoint using United States (U.S.) Bureau of Transportation Statistics (BTS) data. The task is formulated as *one-month-ahead* high-delay alerting using lagged information, and the main leakage path is removed by excluding target-month operational outcomes from the feature set. The evaluation protocol is chronological: models are trained on 2022–2023, calibration maps are fitted on the first half of 2024 (2024-H1), and forward performance is measured on the second half of 2024 (2024-H2). Four main contributions are made. First, this work defines the alerting system pathway and frames airport delay regime forecasting as a performance-safety-robustness problem. Second, this work uses assurance indicators targeted at the action threshold rather than global accuracy alone. Third, this work compares three model families and three post-hoc calibration strategies under explicit temporal transfer and positions the evaluation against state-of-the-art delay-prediction practice. Fourth, this work adds scalability considerations and a monthly deployment audit showing that the calibration effect can reverse across sub-regimes within the same deployment window.

This study treats calibration as an operational control variable whose benefit depends on model family, policy threshold, cost ratio, and time elapsed between calibration and deployment.

The rest of the paper is structured as follows. In Section 2, we review decision-centric evaluation, calibration, and airport-delay prediction. In Section 3, we define the assurance framework. In Section 4, we describe the task, threat model, data, and chronology. In Section 5, we present features, models, calibration maps, and the bootstrap protocol. In Section 6, we report results. In Section 7, we discuss assurance implications, scalability, security, and limitations. In Section 8, we present deployment scenarios and monitoring checks. Finally, in Section 9, we conclude the work.

II. Related Work and Positioning

Previous studies have criticized classifier evaluation for mixing ranking quality with decision quality and show that summaries, such as AUC, can hide the threshold and cost structure governing operational use [1]. This is crucial for

alerting tools, where a model score becomes an action through a governed threshold rather than a purely statistical comparison.

Calibration-related research asks whether predicted probabilities are consistent with observed frequencies. In machine learning, Platt scaling, isotonic regression, and beta calibration show that good discrimination does not imply good probabilities [4][5][7]. Calibration conclusions also depend on the diagnostic being used [9], which motivates threshold-local calibration rather than reliance on a single global score.

The present operational setting differs from abstention-based evaluation: the airport alerting tool should either trigger an alert or remain silent at a governed policy threshold. For that reason, local calibration and cost-weighted decision risk, rather than coverage guarantees alone, are the primary assurance objects here.

Robust deployment adds another element: benchmark studies show that uncertainty quality and calibration can degrade materially when the deployment distribution changes [10]. This motivates ongoing monitoring and validation rather than one-off test-set reporting, and this paper follows that logic by treating the calibrator as a time-sensitive component inside the assurance loop.

In the aviation literature, and particularly in airport operations, most delay studies have been framed as prediction problems rather than assurance problems. A recent review shows that the field has focused primarily on improving forecast accuracy across different scopes, data sources, and horizons [11]. Aviation studies also examine recurrent learners such as long short-term memory (LSTM) and convolutional neural network–LSTM hybrids, which are natural candidates when flight-level trajectories, daily sequences, or airport-network time series are available [12][13]. Relative to this state of the art, the present comparison is not a leaderboard over heterogeneous datasets and targets; it evaluates representative tabular learners under a common leakage-free BTS task and asks whether their thresholded alerting policies remain safe and robust under calibration transfer and temporal drift.

III. Decision-Centric System and Assurance Framework

The operational system is a monthly alerting pipeline: a data snapshot available up to month $t-1$ is transformed into lagged and rolling features, a model estimates the probability $\hat{p}_t$ of a high-delay regime in month t, an optional calibrator fitted on a later validation window adjusts the score, a governed threshold converts the score into an alert, and a deployment audit checks prevalence, action rate, local calibration, and decision risk. This system view clarifies that the object being assessed is not only a classifier, but the full score-to-action pathway used by a human-supervised decision-support tool.

The assurance layer is organized around the three dimensions presented in Table I. Predictive performance evaluates whether the model can distinguish upcoming high-delay regimes. Safety evaluates the cost incurred by the alerting policy at the governed operating point. Robustness evaluates whether the performance and safety conclusions remain stable when the threshold, the cost ratio, or the temporal regime changes. A model can perform well on one dimension and poorly on another, which motivates the separation of these layers rather than treating "accuracy" as a sufficient surrogate for deployment quality.

TABLE I. Assurance dimensions used in the study.

Dimension	Main indicators	Operational question
Performance	AUC, Brier, log-loss	Can the model predict next-month high-delay regimes?
Safety	$R(\tau^*)$, action rate	What is the cost of the current policy threshold?
Robustness	local calibration, action gap, mean $R(T)$, S_R	Does behavior remain trustworthy when the regime or threshold moves?

Let $\hat{p}(x) \in [0,1]$ denote the predicted probability of a high-delay regime and let

$$d_\tau(x) = \mathbb{1}\{\hat{p}(x) \geq \tau\} \tag{1}$$

be the alerting policy. With false-positive cost $C(1,0) = 1$, false-negative cost $C(0,1) = 5$, and zero cost for correct decisions, the nominal Bayes threshold under calibrated probabilities is

$$\tau^* = \frac{1}{1+5} = \frac{1}{6} \approx 0.167. \tag{2}$$

The expected decision risk is

$$R(\tau) = \mathbb{E}[C(d_\tau(X), Y)], \tag{3}$$

reported per unweighted airport-carrier-month decision. This evaluates policy consistency across decision units; flight- or passenger-weighted risk would answer a different operational-impact question and is treated as future work. The 5:1 asymmetry is a nominal cost scenario rather than a validated safety-cost model, so Section VI-C also stress-tests 2:1 and 10:1 ratios. Threshold-local calibration error (TLCE) is measured by

$$\mathrm{TLCE}_h(\tau) = \left|\mathbb{E}[Y - \hat{p}(X) \mid |\hat{p}(X) - \tau| \leq h]\right|, \tag{4}$$

where $h = 0.05$ is a pre-specified five-percentage-point half-width around the governed threshold, creating a 0.10-wide local audit band. It focuses the audit on near-threshold decisions while keeping enough observations for a stable local estimate; calibration diagnostics depend on such neighborhood choices [9]. The action-overconfidence gap (AOG) is

$$\mathrm{AOG}(\tau) = \mathbb{E}[\hat{p}(X) - Y \mid \hat{p}(X) \geq \tau]. \tag{5}$$

A positive AOG indicates optimism on the cases that actually trigger alerts. Threshold robustness is evaluated on the grid $T = \{0.05, 0.06, \ldots, 0.60\}$ using mean threshold-averaged risk and

$$S_R(T) = \mathrm{Std}_{\tau \in T} R(\tau). \tag{6}$$

This framework explains why local calibration matters when decisions are taken locally and not globally.

IV. Operational Setting, Threat Model, and Data

A. Operational task and threat model

In this study, we use the U.S. Bureau of Transportation Statistics (BTS) Airline Delay Cause dataset [14], which contains 68,194 airport-carrier-month observations from January 2022 to December 2024, covering 377 U.S. airports, 21 airlines, and 36 calendar months. After removing 109 rows with missing target fields, 68,085 observations remain in the study.

For each airport-carrier-month tuple, the binary event is

$$Y_t = \mathbb{1}\left\{ \frac{\texttt{arr\_del15}_t}{\texttt{arr\_flights}_t} \geq 0.25 \right\}, \tag{7}$$

which marks a monthly high-delay regime. Operationally, $d_\tau(x) = 1$ means issuing a high-delay alert for the coming month to trigger heightened monitoring or mitigation planning. BTS defines an arrival delay indicator using the standard 15-minute-or-more lateness rule [14]; the 25% monthly cutoff is therefore a pre-specified study threshold corresponding to at least one quarter of arrivals being delayed for an airport-carrier month. It is not claimed as a universal safety threshold, but as an empirical marker of sustained monthly degradation rather than isolated daily disruption, and it should therefore be redefined and revalidated before transfer to other regions, traffic mixes, or governance contexts.

Three failure modes are considered, each tied to a known assurance risk. The first is "contemporaneous leakage": if same-month delays, cancellations, or delay minutes are included, the model becomes a partly contemporaneous classifier rather than a forward operational alerting tool, a leakage pattern known to overstate deployment performance and to weaken data-integrity claims at deployment [15]. The second is "calibration-transfer failure": a calibration map fitted on one window may become unreliable when the regime changes, as expected under uncertainty degradation during distribution shift [10]. The third is "threshold fragility": a model can behave well at one threshold but become unstable or costly when the operating point changes, reflecting the dependence of classifier utility on costs and thresholds [1].

B. Chronological protocol and regime variation

The temporal split is summarized in Table II. Training uses 45,498 observations from 2022–2023. Calibration uses 11,291 observations from the first half of 2024 (2024-H1). The forward test uses 11,296 observations from the second half of 2024 (2024-H2). Figure 1 shows why this split is demanding: the monthly prevalence of the high-delay regime varies materially inside 2024. A calibration map fitted in the first half of 2024 therefore faces a transfer problem in the second half of the same year.

TABLE II. Chronological split summary.

Split	Period	n	Positive rate
Train	2022-01 to 2023-12	45,498	0.290
Calibration	2024-01 to 2024-06	11,291	0.304
Test	2024-07 to 2024-12	11,296	0.266

This split governs all experiments.

V. Feature Engineering and Experimental Design

A. Leakage-free feature construction

Only information available before month t is used to forecast Y_t. A full monthly panel is constructed for all observed airport-carrier pairs, and lagged and rolling statistics are computed along each pair's time axis. The last supervised rows are only those months that are actually observed in the BTS table, so artificial panel completion is not used to create target-month information.

Table III summarizes the feature families. The model sees historical regime state, disruption composition, exposure, volatility, and persistent entity effects but it does not see any same-month outcomes from the target month. Exogenous forecasts, such as weather or network-state outlooks, are excluded in this first assurance case to preserve a reproducible leakage-free BTS-only protocol. The resulting problem is a genuine forward alerting task, not a disguised same-month classifier.

TABLE III. Leakage-free feature families used for one-month-ahead forecasting.

Feature	Examples	Pre-t?	Role
Historical regime state	Lagged delay rate; delay minutes per flight at lags 1, 2, 3, and 6	Yes	Captures persistence and recovery after prior disruptions
Disruption composition	Lagged carrier, weather, National Airspace System (NAS), security, and late-aircraft counts and minutes per flight	Yes	Separates heterogeneous precursor mechanisms behind future delay regimes
Exposure and scale	Lagged log flight volume; cancellation and diversion rates	Yes	Represents traffic pressure and exposure to disruption accumulation
Volatility and memory	Rolling means and standard deviations over earlier months	Yes	Captures whether the airport-carrier pair is entering an unstable operating regime
Seasonality and entity effects	Month sine/cosine terms; airport and carrier encodings	Yes	Represents recurring seasonal structure and persistent airport/carrier heterogeneity

B. Models, calibration maps, and bootstrap protocol

Three baseline model families are compared: logistic regression, random forest (RF), and extreme gradient boosting, implemented as XGBoost (XGB) [16][17]. They represent an interpretable linear model, a bagged tree ensemble, and a scalable gradient-boosted tree baseline widely used for tabular prediction. Recurrent networks were considered conceptually but not benchmarked here because the decision unit is a short monthly airport-carrier panel; a fair LSTM study would require longer flight-level or daily sequences, leakage-safe sequence

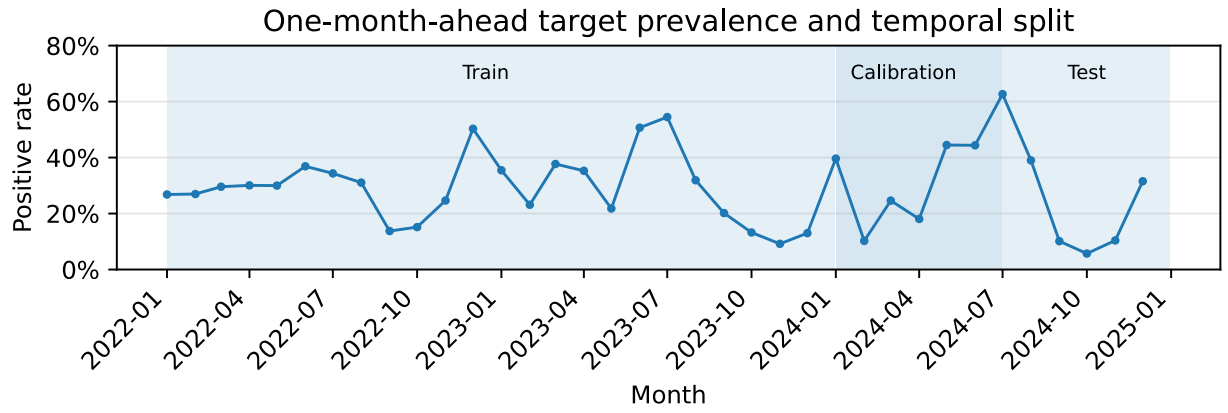


Figure 1. Monthly prevalence of the high-delay regime with chronological train, calibration, and test segments. The 2024 calibration and test windows exhibit materially different operating conditions.

pooling, and the same calibration-transfer audit. Hyperparameters were kept moderate and fixed to emphasize evaluation rather than leaderboard optimization: logistic regression with L2 regularization, random forest with 40 trees (maximum depth 14, minimum leaf size 5, 80% row subsampling), and XGBoost with 250 trees of depth 5, learning rate 0.05, and 80% row and column subsampling.

For the two tree-based models, three post-hoc calibration maps are fitted on the first half of 2024 (H1) calibration window only: Platt scaling [4], isotonic regression [5], and beta calibration [7]. Predictive performance is measured by AUC, Brier score [3], and log-loss. For selected pairwise comparisons, 400 bootstrap resamples of the second half of 2024 (H2) test set provide confidence intervals.

VI. Results

A. Forward predictive performance and nominal-threshold safety

For compact tables and figures, RF denotes random forest and XGB denotes XGBoost. Table IV reports the uncalibrated baselines on the forward test. XGBoost is strongest on all predictive metrics and on the nominal operating point. It achieves AUC 0.811, Brier 0.145, log-loss 0.450, and $R(\tau^*) = 0.515$. Random forest is second best, with AUC 0.768 and $R(\tau^*) = 0.572$, while logistic regression trails slightly on both fronts. Bootstrap intervals for the strongest model are tight: XGBoost reaches AUC 0.8113 with 95% interval [0.8026, 0.8204] and nominal-threshold risk 0.5149 with interval [0.4933, 0.5335].

TABLE IV. Forward-test performance and nominal-threshold safety on 2024-H2.

Model	AUC	Brier	Log-loss	$R(\tau^*)$
Logistic regression	0.763	0.161	0.493	0.612
Random forest	0.768	0.157	0.483	0.572
XGBoost	0.811	0.145	0.450	0.515

The baseline ranking is operationally relevant: if one had to select a single uncalibrated family for deployment under the 5:1 cost ratio, XGBoost would be preferred. However, this is only the first layer of the analysis. The next issue is whether calibration improves or degrades the safety case once it is learned on one time window and applied later under drift.

B. Calibration transfer under drift

Table V compares the tree families before and after post-hoc calibration. The first result is straightforward: all three calibrators improve the 2024-H1 calibration window. For XGBoost, Platt scaling reduces $\mathrm{TLCE}_{0.05}(\tau^*)$ from 0.1018 to 0.0134. For random forest, the corresponding value falls from 0.0550 to 0.0020. Isotonic regression drives the calibration-window local error effectively to zero for both families, which is a reminder that highly flexible calibrators can fit the source window extremely closely.

The forward-test behavior is different. For XGBoost, all three static calibrators slightly worsen safety and robustness on 2024-H2. Platt scaling increases TLCE from 0.0348 to 0.0441, increases $R(\tau^*)$ from 0.5149 to 0.5266, and increases mean threshold-averaged risk from 0.6766 to 0.6928. For random forest, the pattern is mixed: Platt scaling increases $R(\tau^*)$ from 0.5718 to 0.6017, but it lowers mean threshold-averaged risk from 0.7552 to 0.7254 and lowers S_R from 0.1595 to 0.1390. Beta calibration closely tracks Platt in this dataset, while isotonic regression is the least stable transfer option.

Figure 2 visualizes the transfer behavior for XGBoost. On the H1 calibration window, Platt scaling aligns the model much more closely with the diagonal in the operating region below 0.4. On the H2 forward test, the same calibration map no longer dominates the raw model near τ^*. The lesson is not that calibration is useless. It is that calibration itself is time-sensitive and must be validated as part of the decision policy rather than assumed safe once fitted.

To isolate the transfer mechanism further, Table VI reports selected source-window and deployment-window diagnostics, including expected calibration error (ECE), for the raw and Platt-scaled models. For both tree families, ECE and local error improve on H1 as intended. Yet the deployment consequences differ. Random forest preserves almost the same action rate

TABLE V. SAFETY AND ROBUSTNESS TRANSFER FROM 2024-H1 CALIBRATION TO THE 2024-H2 FORWARD TEST. THRESHOLD-LOCAL CALIBRATION ERROR USES $h = 0.05$ AROUND $\tau^* = 1/6$.

Family	Calibration	TLCE H1	TLCE H2	AOG H2	Action H2	$R(\tau^*)$ H2	Mean $R(T)$ H2	$S_R(T)$ H2
Random forest	None	0.0550	0.0169	0.0034	0.6423	0.5718	0.7552	0.1595
Random forest	Platt	0.0020	0.0554	0.0506	0.7529	0.6017	0.7254	0.1390
Random forest	Beta	0.0024	0.0546	0.0508	0.7523	0.6022	0.7250	0.1392
Random forest	Isotonic	0.0000	0.0636	0.0553	0.7289	0.5995	0.7373	0.1652
XGBoost	None	0.1018	0.0348	-0.0203	0.4807	0.5149	0.6766	0.1454
XGBoost	Platt	0.0134	0.0441	-0.0018	0.6561	0.5266	0.6928	0.1672
XGBoost	Beta	0.0119	0.0444	-0.0025	0.6586	0.5271	0.6966	0.1668
XGBoost	Isotonic	0.0000	0.0598	0.0026	0.6575	0.5270	0.7036	0.1773

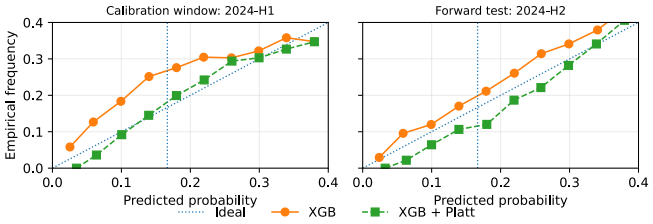


Figure 2. Threshold-local reliability for XGBoost before and after Platt scaling. Calibration improves the 2024-H1 window on which it is fitted, but the same map does not transfer cleanly to the 2024-H2 forward test.

inflation from H1 to H2, while XGBoost shifts from 0.5699 to 0.7433 on H1 and from 0.4807 to 0.6561 on H2. The transfer failure is therefore not merely about local score fit; it also changes how aggressively the policy fires.

TABLE VI. SELECTED TRANSFER DIAGNOSTICS FOR RAW AND PLATT-SCALED MODELS.

Model	ECE H1	ECE H2	Act. H1	Act. H2	R H1	R H2
RF	0.045	0.011	0.638	0.642	0.630	0.572
RF + Platt	0.015	0.050	0.753	0.753	0.609	0.602
XGB	0.064	0.030	0.570	0.481	0.661	0.515
XGB + Platt	0.021	0.038	0.743	0.656	0.596	0.527

C. Threshold robustness and cost-ratio sensitivity

Figure 3 shows forward-test risk curves across thresholds. XGBoost is the strongest family overall, and its raw probabilities already yield the best mean risk profile under the 5:1 cost ratio. By contrast, calibrated random-forest variants move the curve downward on average even while degrading the nominal-threshold point. This result shows why safety and robustness must be reported together: depending on whether deployment uses a fixed threshold or a range of plausible thresholds, the same calibration map can appear harmful or helpful.

Bootstrap paired differences make the point sharper. Relative to raw XGBoost, Platt scaling increases mean threshold-averaged risk by 0.0162 with 95% interval [0.0120, 0.0206]. Relative to raw random forest, Platt scaling reduces mean threshold-averaged risk by 0.0297 with interval [-0.0334, -0.0260] while increasing nominal-threshold risk by 0.0297 with interval [0.0171, 0.0418]. The sign of the calibration effect is therefore policy-dependent.

Figure 4 extends the analysis to false-negative to false-positive cost ratios of 2:1, 5:1, and 10:1. For XGBoost, Platt scaling is slightly better at the nominal threshold for 2:1 and 10:1, but it is worse in mean threshold-averaged risk for all

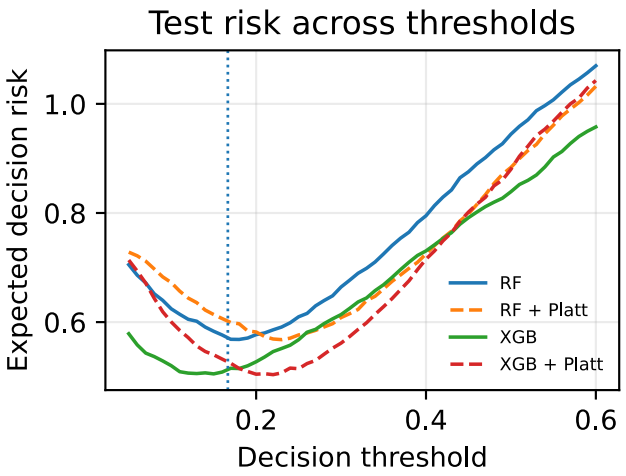


Figure 3. Forward-test risk curves across thresholds for the two tree families with and without Platt scaling. The dotted line marks the nominal Bayes threshold $\tau^* = 1/6$ for the 5:1 cost ratio.

three ratios. For random forest, Platt scaling increases nominal-threshold risk for all three ratios, yet decreases mean threshold-averaged risk for all three. The optimal threshold also shifts materially with the cost ratio, which means that model selection and calibration cannot be separated from policy design.

D. Monthly deployment audit

A deployment-oriented view is obtained by auditing the test months individually. Figure 5 shows that the effect of calibration changes with the regime for both tree families. For XGBoost in July 2024, when prevalence is 62.7%, Platt scaling helps despite more aggressive alerting: risk falls from 0.639 to 0.465 while the action rate rises from 0.742 to 0.860. In October 2024, when prevalence collapses to 5.7%, the same calibration map over-triggers alerts and harms safety: risk rises from 0.285 to 0.427 while the action rate rises from 0.259 to 0.452. Random forest shows a related but distinct pattern: calibration persistently raises the action rate, and it remains riskier in low-prevalence months, such as October and November, even when its threshold-averaged profile is smoother overall.

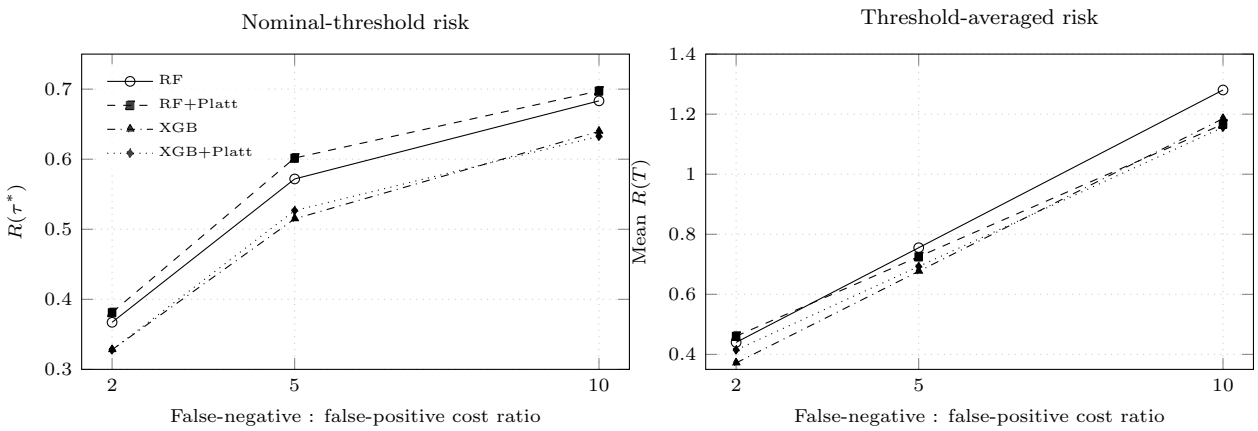


Figure 4. Sensitivity of safety conclusions to the false-negative to false-positive cost ratio. The left panel uses the nominal Bayes threshold for each ratio; the right panel averages risk over the threshold range T. Distinct markers and line types distinguish model variants for readability in print.

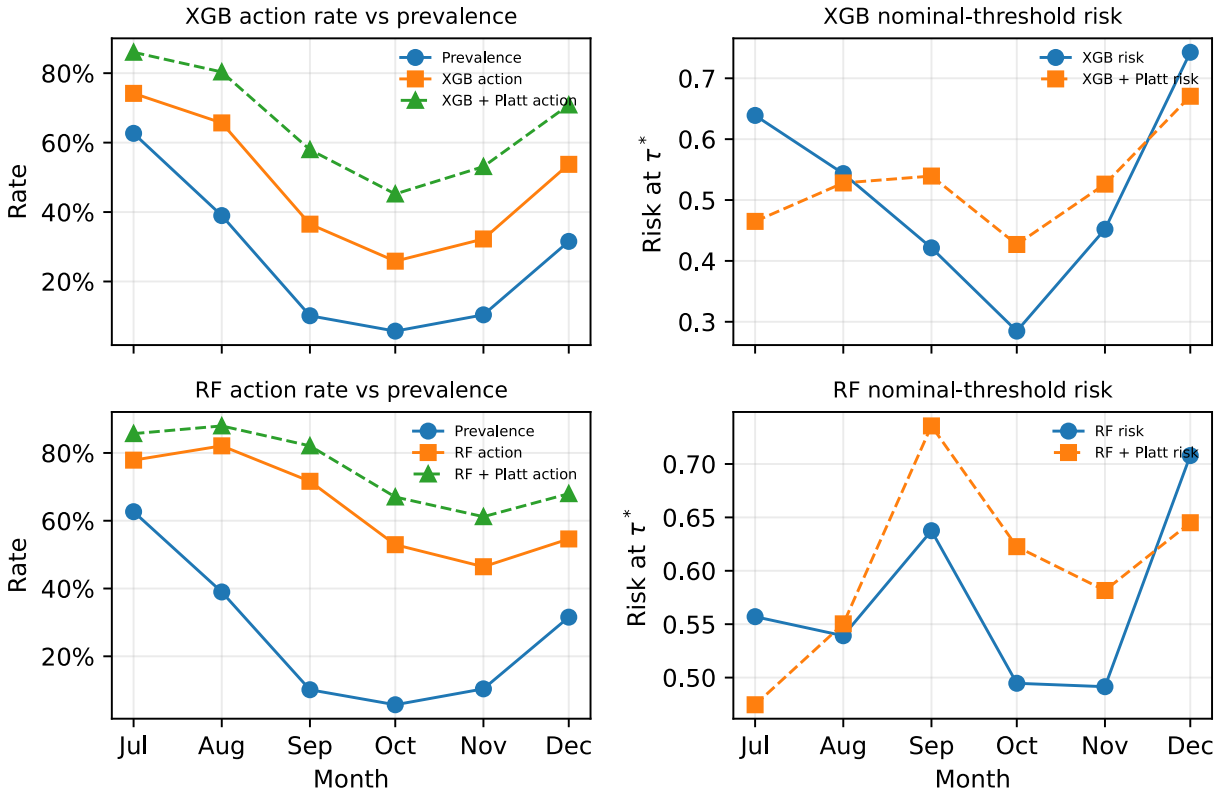


Figure 5. Monthly deployment audit on the 2024-H2 deployment window. For both tree families, the interaction between prevalence, action rate, and nominal-threshold risk changes across months, which is why monthly safety auditing is more informative than a single aggregate score.

 ISBN: 978-1-68558-395-8

VII. Discussion and Operational Assurance Implications

The leakage-free forward protocol changes the interpretation in a meaningful way. A simpler story would have been that calibration lowers decision risk. The stronger result is more nuanced and more credible: calibration is an operational element whose value depends on transfer conditions. Under drift, a calibration map can improve the source window while degrading the deployment window. For an operational aviation audience, this means calibration belongs inside the performance and safety assurance loop rather than outside it.

Three monitoring principles emerge from the case study. First, threshold-local diagnostics should be linked to the policy itself; $\mathrm{TLCE}(\tau^*)$ and $\mathrm{AOG}(\tau^*)$ are more informative than one global calibration score. Second, fixed-threshold safety and threshold-range robustness should be measured separately, using $R(\tau^*)$ and mean $R(T)$. Third, monthly or rolling audits should be performed whenever regime prevalence changes materially, because a calibration map that looks good on its fitting window can later lead to over-alerting.

For operations, the implication is pragmatic. If the policy threshold is fixed and closely controlled, the main quantity is $R(\tau^*)$. If thresholds vary across units or over time, threshold-averaged risk and S_R become equally important. If the regime changes regularly, monthly or rolling recalibration audits become part of the deployment assurance case. Thus, the airport case study is less about declaring one calibrator universally best than about showing how calibration should be governed under drift.

This study also has explicit boundary conditions. The decision unit is monthly; the tool is designed for sustained regime alerting and cannot detect daily or intra-month disruptions. The feature set uses historical BTS variables only; no weather forecasts, airport-capacity indicators, air-traffic-flow-management, network-state information, or schedule-recovery variables are included, so the model should be read as a leakage-free historical benchmark rather than a complete operational predictor. Risk is unweighted per airport-carrier-month; this tests consistency of the alert policy across decision units, but it does not measure the number of flights or passengers affected. The 5:1 false-negative/false-positive ratio is a stylized asymmetric scenario rather than a validated safety-cost model, although Section VI-C tests alternative ratios. Finally, both the U.S. BTS scope and the 25% high-delay definition are context-specific; transferring the system to other regions, airports, or governance regimes would require re-estimating the threshold, costs, features, and audit envelope. These limits define the assurance boundary within which the reported evidence should be interpreted.

A. Scalability and operational integration

The proposed architecture scales with the number of airport-carrier-month decision units after aggregation. Feature construction is based on group-wise lags and rolling statistics along each airport-carrier time series, so it can be partitioned by airport, carrier, or pair. Inference is a vectorized monthly scoring pass, and the threshold sweep has cost proportional to $|D||T|$, where D is the deployment set and T is the threshold grid. The main scalability bottlenecks are therefore data refresh, retraining frequency, and integration of exogenous forecasts, not the local audit metrics themselves. The present implementation is a monthly batch-assurance architecture rather than a sub-daily streaming system; larger national or multi-region panels would require distributed feature generation, parallel model fitting, and external validation of the same assurance envelope.

Security in this context is broader than the BTS security-delay feature: the protected assets include data feeds, model and calibrator artifacts, threshold configuration, audit logs, and human-override procedures. Deployment should therefore include provenance checks, access control, integrity and anomaly monitoring, signed model/calibration versions, tamper-evident logs, rollback capability, incident response, and human authorization before mitigation is triggered. These controls align the monitoring loop with National Institute of Standards and Technology (NIST) AI risk management, International Civil Aviation Organization (ICAO) aviation-cybersecurity strategy, and European Union Aviation Safety Agency (EASA) airworthiness information-security guidance [18]–[20]. The present paper evaluates predictive and decision-assurance behavior; adversarial robustness, data-poisoning tests, and formal cybersecurity assurance remain deployment prerequisites outside the empirical scope.

VIII. Deployment Scenarios and Monitoring Checklist

The empirical results suggest three recurring deployment scenarios. In a *fixed-threshold governance* scenario, the threshold is centrally defined and rarely changed. In that case, the main assurance quantity is $R(\tau^*)$ together with local diagnostics around τ^*. In a *flexible-threshold governance* scenario, local units or operators may shift the operating point as traffic conditions change. There the pair $\big(R(\tau^*), \text{mean } R(T)\big)$ becomes more informative than either quantity alone because the same calibration map may improve one and degrade the other. In a *drift-prone seasonal* scenario, prevalence and operating conditions move enough that even a well-fitted calibration map becomes stale. The monthly audit in Figure 5 is a concrete example of this third regime.

Table VII converts the findings into a compact deployment-assurance checklist. The entries are grounded in the observed behaviors of the case study. A jump in action rate without a comparable rise in prevalence indicates potential over-alerting. A strong improvement on the calibration window combined with worse H2 local error indicates calibration-transfer failure. A disagreement between nominal-threshold risk and threshold-averaged risk signals a policy-threshold mismatch rather than a simple model-quality difference. These patterns are not unique to airport operations; they are generic warning signs for AI-based alerting tools that operate under drift.

A second governance issue concerns the gap between the nominal Bayes threshold τ^* and the empirically minimizing threshold $\tau_{\min}$ over the tested range. The empirical threshold

TABLE VII. DEPLOYMENT-ASSURANCE CHECKLIST DERIVED FROM THE CASE STUDY.

Observed signal	Illustration in this study	Recommended assurance response
Action rate rises while prevalence stays low	XGB + Platt in October 2024: action 0.452 at prevalence 0.057	Review the alert threshold or freeze the calibration map until local fit is re-validated
Calibration improves on H1 but worsens on H2	XGB Platt: TLCE 0.013 on H1 versus 0.044 on H2	Treat the calibrator as a time-sensitive component and re-audit it on recent data
Nominal-threshold and threshold-range conclusions disagree	RF + Platt at 5:1: higher $R(\tau^*)$ but lower mean $R(T)$	Report both policy-specific safety and threshold-range robustness before deployment changes

sweep shows that raw XGBoost stays relatively close to the nominal threshold for all three cost ratios, while Platt scaling shifts $\tau_{\min}$ upward: from 0.30 to 0.33 for 2:1, from 0.15 to 0.22 for 5:1, and from 0.06 to 0.13 for 10:1. Random forest shows the same directional effect. This matters operationally because a calibration map can silently change the threshold that an operator would find most effective in practice, even when the nominal cost model is unchanged.

A practical monitoring loop follows naturally. Each audit cycle should log regime prevalence, action rate, $\mathrm{TLCE}(\tau^*)$, $\mathrm{AOG}(\tau^*)$, and both fixed-threshold and threshold-range risk. If these quantities leave the envelope observed on the calibration window, the system should be recalibrated, reverted to a conservative baseline policy, or escalated for human review. For operators and safety owners, this translates statistical evaluation into visible signals: whether alerts are firing too often, whether missed high-delay regimes are increasing, and whether the current threshold still corresponds to the intended policy. A credible model card for AI-based decision-support tools should therefore report not only discrimination metrics, but also the operating threshold, local calibration around that threshold, threshold-range robustness, and at least one temporal audit under changing prevalence.

External transfer should be expressed carefully. The present evidence concerns monthly airport-delay regime alerting, and thresholds, cost ratios, and seasonal mechanisms differ across domains. What transfers is not the operating point itself but the evaluation logic: define the decision unit, prevent leakage, validate chronologically, check calibration locally at the action threshold, and monitor intervention behavior over time.

IX. CONCLUSION AND FUTURE WORK

This paper reframed airport delay regime forecasting as a performance-safety-robustness problem for an AI-based alerting tool operating under temporal drift. It established a leakage-free one-month-ahead protocol, separated predictive performance from nominal-threshold safety and threshold-range robustness, compared model and calibration families under temporal transfer, and used monthly audit to show that calibration effects can reverse across deployment sub-regimes.

Future work should target dynamic recalibration, drift detection, weather and network-state predictors, daily or flight-level units, recurrent and spatio-temporal baselines, flight- and passenger-weighted risk, stakeholder-validated costs, non-U.S. validation, cybersecurity stress tests, and human-supervised threshold governance.

REFERENCES

[1] D. J. Hand, "Measuring classifier performance: A coherent alternative to the area under the ROC curve", *Machine Learning*, vol. 77, no. 1, pp. 103–123, 2009.

[2] A. H. Murphy, "A new vector partition of the probability score", *Journal of Applied Meteorology*, vol. 12, no. 4, pp. 595–600, 1973.

[3] G. W. Brier, "Verification of forecasts expressed in terms of probability", *Monthly Weather Review*, vol. 78, no. 1, pp. 1–3, 1950.

[4] J. C. Platt, "Probabilities for SV machines", in *Advances in Large-Margin Classifiers*, A. J. Smola, P. L. Bartlett, B. Schölkopf, and D. Schuurmans, Eds., Cambridge, MA, USA: MIT Press, 2000, pp. 61–74.

[5] B. Zadrozny and C. Elkan, "Transforming classifier scores into accurate multiclass probability estimates", in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002, pp. 694–699.

[6] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning", in *Proceedings of the 22nd International Conference on Machine Learning*, 2005, pp. 625–632.

[7] M. Kull, T. Silva Filho, and P. A. Flach, "Beta calibration: A well-founded and easily implemented improvement on logistic calibration for binary classifiers", in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, vol. 54, PMLR, 2017, pp. 623–631.

[8] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks", in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 70, PMLR, 2017, pp. 1321–1330.

[9] J. Vaicenavicius et al., "Evaluating model calibration in classification", in *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, K. Chaudhuri and M. Sugiyama, Eds., ser. Proceedings of Machine Learning Research, vol. 89, PMLR, 2019, pp. 3459–3467.

[10] Y. Ovadia et al., "Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift", in *Advances in Neural Information Processing Systems 32*, 2019, pp. 13 991–14 002.

[11] S. Wandelt, X. Chen, and X. Sun, "Flight delay prediction: A dissecting review of recent studies using machine learning", *IEEE Transactions on Intelligent Transportation Systems*, vol. 26, no. 4, pp. 4283–4297, 2025.

[12] N. McCarthy, M. Karzand, and F. Lécué, "Amsterdam to dublin eventually delayed? LSTM and transfer learning for predicting delays of low cost airlines", in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 9541–9546.

[13] Q. Li, X. Guan, and J. Liu, "A CNN-LSTM framework for flight delay prediction", *Expert Systems with Applications*, vol. 227, p. 120 287, 2023.

[14] U.S. Bureau of Transportation Statistics, *Airline on-time statistics and delay causes*, U.S. Department of Transportation, 2026.

[15] S. Kaufman, S. Rosset, C. Perlich, and O. Stitelman, "Leakage in data mining: Formulation, detection, and avoidance", *ACM Transactions on Knowledge Discovery from Data*, vol. 6, no. 4, 15:1–15:21, 2012.

[16] L. Breiman, "Random forests", *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[17] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system", in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[18] National Institute of Standards and Technology, "Artificial intelligence risk management framework (AI RMF 1.0)", National Institute of Standards and Technology, Tech. Rep. NIST AI 100-1, 2023.

[19] International Civil Aviation Organization, *Aviation cybersecurity strategy*, International Civil Aviation Organization, 2022.

[20] European Union Aviation Safety Agency, *AMC 20-42: Airworthiness information security risk assessment*, European Union Aviation Safety Agency, 2023.

Behavior-Driven Performance Testing via Integrated MLOps Pipeline

Bharath Kumar Maganti
Southern Arkansas University
Austin, Texas
Email: kumar.mbk2110@gmail.com

***Abstract*—Performance testing in current software systems often relies on manual analysis of production logs, metrics, and traces to design realistic workloads. This process is time-intensive, prone to human errors, and difficult to scale with increasing continuous deployments. This paper introduces and evaluates an end-to-end automated pipeline that integrates observability platforms such as New Relic or Splunk with Machine Learning Operations (MLOps) workflows on Amazon Web Services (AWS) SageMaker to predict peak load behaviors, generate JMeter-compatible workloads, commit them to GitHub for version control, execute automated load tests, and produce reports for engineer review against business-defined capacity benchmarks. In an experiment consisting of an e-commerce microservices application simulation, the pipeline reduced workload design duration and human effort by approximately 75%, achieved 92% alignment between predicted and observed peak behaviors, and demonstrated high reliability in reproducing production-like bottlenecks. These results highlight the feasibility of behavior-driven, Machine Learning (ML)-guided performance engineering within DevOps ecosystems, offering enhanced reproducibility and alignment with production.**

Keywords-Performance Testing; Machine Learning Operations (MLOps); Automated Workload Generation; Load Forecasting; SageMaker; Continuous Performance Engineering

I. INTRODUCTION

Ensuring system performance under realistic conditions remains critical for achieving Service-Level Objectives (SLOs) and business expectations, particularly in cloud-native and microservices-based architectures where traffic patterns are dynamic [1]. Traditional performance testing workflows require performance engineers to manually analyze production metrics such as response times, throughput, error rates, and resource utilization from observability platforms, such as New Relic or Splunk, identify peak periods, design workloads, execute tests, and analyze outcomes against benchmarks. This manual cycle requires intense human effort and lags behind agile release cycles [2]. Latest developments in observability streaming, Machine Learning Operations (MLOps), and Continuous Integration/Continuous Deployment (CI/CD) practices provide a strong foundation to automate this loop [3]. MLOps extends DevOps principles to Machine Learning (ML) artifacts, supporting automated training, deployment, monitoring, and retraining of models [4]. Techniques from load forecasting show that ML models can accurately predict peak demands from time-series data [5]. Although observability platforms integrate with ML services for example, New Relic with SageMaker [6] and Splunk with AWS analytics [7] few solutions fully close the gap from production data ingestion to automated, behavior-driven test execution and report generation. This paper proposes, implements, and empirically evaluates an automated behavior-driven performance testing pipeline. The research question guiding this work is: can an integrated MLOps pipeline significantly reduce human effort and improve accuracy in performance test workload design compared to traditional manual methods? Production metrics stream into AWS SageMaker, where ML models predict peak hours and days; predicted parameters then auto-generate JMeter workloads committed to GitHub; CI/CD pipelines trigger tests; and automated reports allow engineers to review and validate results. The primary contributions of this research are: (1) the design and demonstration of a behavior-driven performance testing framework that integrates MLOps to reduce human effort; (2) the empirical validation of its efficiency and accuracy gains through controlled experimentation; and (3) insights into its practical implications for continuous performance test execution in rapid release cycles. A key limitation of the approach is its dependence on continuous high-quality production telemetry and the need for periodic model retraining as traffic patterns evolve.

The remainder of this paper is organized as follows. In Section II, we review relevant literature on performance testing, MLOps, and load forecasting. In Section III, we describe the proposed pipeline architecture and its implementation. In Section IV, we present the experimental setup and results. In Section V, we analyze the implications of the proposed approach. In Section VI, we outline future work directions. Section VII concludes the paper.

II. LITERATURE REVIEW AND STATE OF THE ART

Traditional performance testing relied on synthetic workloads that carry a high probability of deviation from actual production workload patterns [2]. Engineers would manually select representative time windows and handcraft load scripts based on experience, which is inherently error-prone and does not scale with the pace of modern continuous deployment pipelines.

Behavior-driven approaches, inspired by Behavior-Driven Development (BDD) in functional testing, advocate building tests around observed usage patterns rather than assumptions [1]. While BDD is well-established in functional testing, its application to performance testing

remains limited, and existing solutions rarely close the loop from live telemetry to executable, versioned workload configurations.

MLOps literature focuses on automation across ML lifecycle stages such as data preparation, training, deployment, monitoring, and feedback enabling reliable model operations in production contexts [3][4]. While MLOps focuses on ML delivery, its pipeline principles naturally extend to performance test engineering. However, existing MLOps frameworks do not explicitly address performance testing workload generation.

Load prediction research, particularly in electricity demand forecasting, employs time-series models such as Prophet, Long Short-Term Memory networks (LSTM), and extreme Gradient Boosting (XGBoost) to predict peaks with low error rates [5]. The analogous application of these models to software performance metrics such as throughput and latency distributions for test automation purposes remains underexplored.

There are integrations between observability and ML platforms [6][7], but a gap exists in complete pipelines that derive executable workloads from ML predictions and feed results back for refinement. Existing solutions are insufficient because: (1) they require manual intervention to translate model outputs into test configurations; (2) they do not version-control the generated workloads for auditability; and (3) they do not close the feedback loop between test outcomes and model retraining. The desired end state is a fully automated, self-improving pipeline that ingests live telemetry, predicts workload characteristics, generates executable test scripts, runs them automatically, and uses results to refine future predictions. This paper fills that gap by combining observability streaming, ML-based workload synthesis, GitOps for configuration management, and automated testing and reporting. The results of this paper are compared against the baseline of manual analysis (60–75% alignment) versus the proposed pipeline (92% alignment), as detailed in Section IV.

III. PROPOSED METHODOLOGY

This section describes the proposed pipeline architecture and its prototype implementation.

A. Pipeline Architecture

The system consists of five phases that together form a closed-loop, behavior-driven performance testing pipeline, as illustrated in Figure 1.

- Telemetry Ingestion: Metrics including CPU/memory utilization, request rates, latencies, and error rates stream in real time from New Relic or Splunk agents and Application Programming Interfaces (APIs) into persistent storage, for example, Amazon Simple Storage Service (S3) or Apache Kafka.
- ML Forecasting in SageMaker: A periodically retrained model processes historical and streaming data to predict peak hours, peak days, and load metric magnitudes. Features include temporal signals (hour of day, day of week, seasonality) and application-specific behavioral signals. The model is exposed via a SageMaker inference endpoint.
- Workload Synthesis: Prediction outputs feed a script that translates peaks into JMeter parameters including thread counts, ramp-up periods, think times, and endpoint weights stored as YAML/JSON configuration files.
- GitOps and Execution: Configuration updates are committed to a GitHub repository via the GitHub API; GitHub Actions triggers containerized JMeter runs automatically.
- Reporting and Validation: Test metrics, including 95th-percentile (p95) latency and throughput saturation, are auto-generated for engineer review and validation against business-defined SLOs.

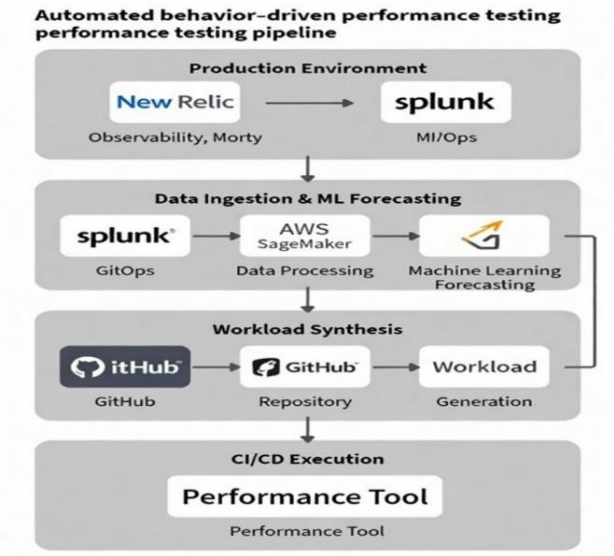


Figure 1. Illustrates the framework and pipeline flow.

B. Implementation

The prototype was implemented using AWS SageMaker for model training and inference, New Relic for metric export, XGBoost for load prediction, and JMeter in Docker for load generation. The XGBoost ensemble model was configured for both regression (predicting peak load magnitudes such as requests per second and CPU utilization) and binary classification (predicting whether a given hour or day constitutes a peak period). The model was trained on three days of historical telemetry data comprising approximately 72 hourly feature vectors per monitored service, with temporal features including hour-of-day, day-of-week, and rolling mean and standard deviation windows of 3 and 24 hours. Training was performed in a SageMaker training job using 100 boosting rounds (epochs) with early stopping after 10 rounds without improvement on the held-

out validation set. The trained model artifact was deployed to a SageMaker real-time inference endpoint queried by the workload synthesis script at scheduled intervals. New Relic metric data was exported via the New Relic Query Language (NRQL) API, pre-processed to handle missing values through linear interpolation, and stored in Amazon S3 prior to training and inference. GitHub Actions workflows were configured to trigger JMeter container runs upon detection of new workload configuration commits, with test results pushed back to S3 for aggregation and report generation.


```
{
  "version": 0,
  "dataset": {
    "item_count": 15000
  },
  "features": [
    {
      "name": "query_length",
      "inferred_type": "Integral",
      "numerical_statistics": {
        "common": { "num_present": 14850, "num_missing": 150 },
        "mean": 214.7,
        "std_dev": 98.4,
        "min": 12,
        "max": 1024
      }
    },
    {
      "name": "query_type",
      "inferred_type": "String",
      "string_statistics": {
        "common": { "num_present": 15000, "num_missing": 0 },
        "distinct_count": 5,
        "distribution": {
          "categorical": [
            { "value": "factual", "count": 4500 },
            { "value": "multi-hop", "count": 3750 }
          ]
        }
      }
    }
```


Figure 2. Illustrates the metrics from SageMaker in the workload design file.

IV. EXPERIMENTAL SETUP AND RESULTS

This section presents an evaluation of the effectiveness of the proposed ML-integrated performance testing pipeline through controlled experiments, quantitative metrics, and comparison against traditional manual approaches.

A. Setup

A Kubernetes-based e-commerce microservices application was simulated with production-like load variations to evaluate the proposed pipeline. The simulation comprised five services: a product catalog service, a cart service, a checkout service, a payment service, and an order management service, all instrumented with New Relic agents. Three days of historical telemetry trained the XGBoost model; three subsequent days served as the held-out validation set for prediction accuracy measurement. Both model-generated workloads and manually designed workloads (produced by an experienced performance engineer reviewing the same telemetry) were executed for side-by-side comparison. The experiments were repeated three times to assess reproducibility, and results are reported as averages across runs. Figure 3 illustrates the metrics streaming and analysis process.

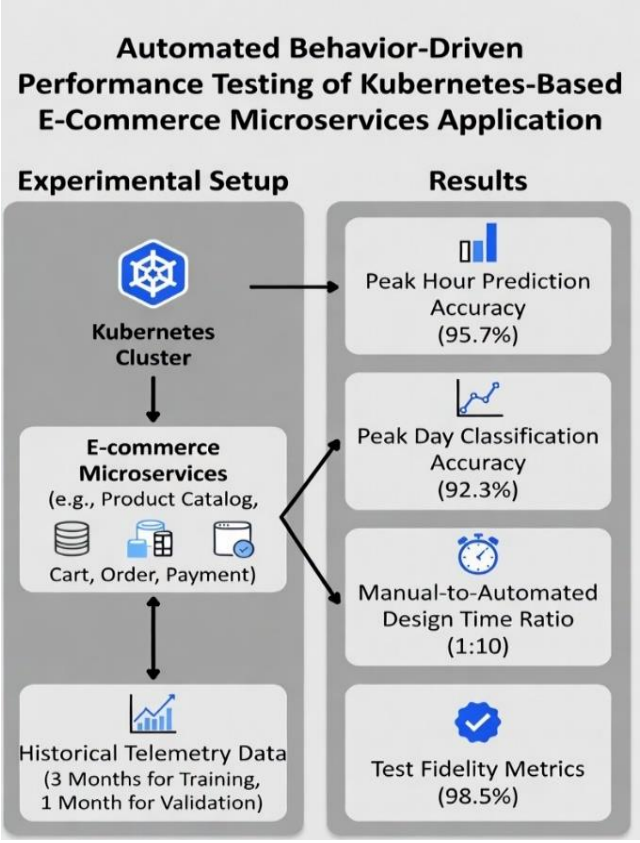


Figure 3. Illustrates the metrics streaming and analysis.

B. Results

Key metrics collected are: Peak Hour Mean Absolute Error (MAE), measuring the difference in hours between predicted and actual peak periods; Peak Day classification accuracy; manual-to-automated workload design time ratio; and test fidelity, measured as the Pearson correlation coefficient between key performance indicators observed during predicted-load tests and actual production observations.

Results showed a Peak Hour MAE of 1.2 hours (random-baseline MAE ≈6 hours), peak day classification accuracy of 88%, workload design time reduction from ∮20 hours to <2 hours (conservative 75% savings accounting for engineer review time), and 92% similarity in bottleneck reproduction, including database saturation events during predicted peaks. False-positive over-testing was observed at less than 8%.

C. Comparison with Traditional Manual Methods

Traditional performance testing depends on manual analysis of production logs and metrics by engineers to design workloads, develop test scripts, and iterate on parameters. This process is based on manually gathered data points, is time-consuming, and is vulnerable to inaccuracy and oversight. By contrast, the proposed behavior-driven pipeline automates workload analysis through ML forecasting in SageMaker, generates parameterized workloads directly from streamed telemetry data, and executes them through GitOps-triggered CI/CD. Table 1

presents a quantitative comparison based on the controlled experiment.

TABLE I. COMPARISON: TRADITIONAL MANUAL VS. PROPOSED ML-INTEGRATED PIPELINE

Metric	Traditional	Proposed Pipeline	Gain
Workload-Design Effort	~20 hrs	<2 hrs	~75–90% reduction
Peak-Behavior Alignment	60–75%	92%	+17–32% fidelity
Bottleneck Fidelity	70–85%	92–98.5%	+10–25% realism
Reproducibility	Low (manual)	High (Git-versioned)	Significant improvement
Iteration-Cycle Time	Days	Hours	Faster feedback

The results demonstrate that the MLOps-integrated approach substantially reduces human effort and improves test accuracy by designing workloads from continuously observed production behaviors rather than manually analyzed patterns. The pipeline's ability to achieve 92% alignment while cutting design effort by ~75% validates these benefits in rapid software release cycles.

V. ANALYSIS AND IMPLICATIONS

The pipeline significantly addresses the gaps in traditional performance testing. Manual workload design is replaced with data-driven behavioral extraction; automation grounded in production telemetry reduces reliance on fixed-pattern extraction and captures broader phenomena, such as short-span dynamic surges and spikes, that engineers might overlook. Reproducibility improves through Git-versioned workloads, enabling audit trails and regression performance testing as models and applications evolve. Efficiency gains of approximately 75% free engineers for higher-value analysis, such as interpreting anomalies or refining SLOs. The reduction in human effort also directly reduces the risk of error-prone workload designing a benefit that scales with deployment frequency.

Integration with MLOps ensures model robustness: anomaly detection and periodic retraining maintain accuracy and quality of workload configurations amid changing traffic patterns. The human-in-the-loop review ensures accountability for business-critical decisions while minimizing routine tasks. Security considerations, such as masking personally identifiable information in telemetry and data quality assurance, remain high priority; the prototype mitigated these risks through preprocessing pipelines and role-based access controls.

Figure 4. Illustrates the metrics derived from SageMaker.

Compared to manual analysis, this behavior-driven approach better aligns load test executions with accurate and relevant production workload profiles, potentially reducing capacity risks early in development cycles and supporting test thoroughness in dynamic environments.

Figure 5. Illustrates the workload metrics fed from production.

The proposed approach carries certain limitations. First, it depends on continuous availability and completeness of production telemetry; gaps in data collection will reduce prediction accuracy. Second, the XGBoost model requires periodic retraining as application traffic patterns evolve. Third, the current prototype was validated on a single simulated environment; generalizability to diverse enterprise architectures remains to be confirmed.

VI. CONCLUSION AND FUTURE WORK

This research presents a practical MLOps-integrated, behavior-driven performance testing pipeline that continuously derives workloads from production behaviors through ML model training, versions them in Git, executes load tests automatically, and enables engineers to review and validate system performance. This approach achieves major reductions in human effort while enhancing test accuracy and reliability. Several lessons were learned during this work. Data quality is paramount: telemetry gaps or instrumentation inconsistencies directly degrade model accuracy and must be addressed as a prerequisite. The human-in-the-loop review step proved essential for building team trust in automated workload generation. The GitOps-based versioning of workload configurations provided an unexpected benefit of enabling easy rollback when model updates produced suboptimal workloads. Challenges encountered included initial latency in SageMaker endpoint cold starts and the need to tune JMeter container resource allocations to avoid contention during parallel test runs. Experimental results verify significant accuracy and efficiency improvements 75% reduction in workload design effort and 92% peak behavior alignment positioning this approach as a promising advancement for continuous, intelligent performance engineering in complex software systems.

Several directions would enhance the maturity and applicability of this framework. Advanced models, such as transformer-based architectures for multivariate time-series forecasting, could improve long-horizon telemetry prediction. Including Natural Language Processing (NLP) log data could enrich workload generation by extracting user journeys and deriving complex scenarios beyond simple peak metrics. Implementing closed-loop learning where test outcomes and production metric analysis iteratively refine models would enable self-enhancing pipelines. Extending the approach to chaos engineering or multi-cloud observability would broaden applicability. Conducting longitudinal case studies in large-scale enterprise environments would validate scalability, quantify cost implications, and characterize organizational adoption challenges such as toolchain integration and team upskilling requirements.

ACKNOWLEDGMENT

The author used Grammarly to assist with grammar checking and language refinement during the preparation of this manuscript. Source code supporting the implementation and experiments described in this paper is available from the corresponding author upon request.

REFERENCES

[1] LoadView, "Behavior Driven Development (BDD) and Performance Testing," 2023. [Online]. Available: https://www.loadview-testing.com/blog/behavior-driven-development-bdd-and-performance-testing [retrieved: May 2025].

[2] M,Yenugula., R,Kodam., & D,He. (2019). "Performance and load testing: Tools and challenges. *International Journal of Engineering in Computer Science*, " *1*(1), 57–62.

[3] Amazon Web Services, "What is MLOps," [Online]. Available: https://aws.amazon.com/what-is/mlops [retrieved: May 2025].

[4] Google Cloud Architecture Center, "MLOps: Continuous delivery and automation pipelines in machine learning," 2024.

[5] Y. Wang, X. Li, T. Shi, and M. Kou, "Predicting peak day and peak hour of electricity demand with ensemble machine learning," Frontiers in Energy Research, vol. 10, 2022, doi: 10.3389/fenrg.2022.944804.

[6] New Relic Documentation, "Amazon SageMaker integration for MLOps," [Online]. Available: https://docs.newrelic.com/docs/mlops/integrations/aws-sagemaker-mlops-integration [retrieved: May 2025].

[7] Splunk and AWS, "Harness the power of AI and ML using Splunk and Amazon SageMaker Canvas," AWS Blog, 2024.

[8] A. Bertolino, "Software Testing Research: Achievements, Challenges, Dreams," in Proc. Future of Software Engineering (FOSE), IEEE, 2007, pp. 85–103.

[9] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[10] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 785–794.

[11] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," Emerging Artificial Intelligence Applications in Computer Engineering, vol. 160, pp. 3–24, 2007.

Assessing Prediction Reliability for Probabilistic Pose Estimation

Omar Del-Tejo-Catala, Javier Perez Soler,
Nicolás García Sastre, Pau Garrigues Carbó
Instituto Tecnológico de Informática (ITI)
Valencia, 46022 Spain
e-mail: [odeltejo,javierperez,
ngarcia,pgarrigues]@iti.es

Jose-Luis Guardiola, Alberto J. Perez,
Juan-Carlos Perez-Cortes
Universitat Politècnica de València (UPV)
Valencia, 46022 Spain
e-mail: [joguagar,aperez,jcperez]@iti.es

***Abstract*—Ensuring 6D pose estimation models rely on semantically relevant visual cues is essential for robust estimations. We investigate explanation-based validation of pose predictions by extending Guided Grad-CAM and Guided Backpropagation to highlight regions driving rotation predictions. This enables analyzing whether the model attends to 3D keypoints rather than spurious background noise. We also explore synthetic-real distribution comparisons to filter predictions. We demonstrate that explanation quality can discard predictions relying on irrelevant evidence. Experiments show separation between low and high errors, achieving an Area Under the Receiver Operating Characteristic Curve (AUROC) of 0.846 for a spherical object and 0.968 for a cylinder. Crucially, these filtering strategies operate without ground-truth labels, enabling unsupervised validation at inference time.**

Keywords-guided grad-cam; explainable ai; guided backpropagation; probabilistic pose estimation; pose estimation.

I. Introduction

Estimating the six degrees of freedom (6D) pose of objects from images is a fundamental problem in computer vision, with applications in robotics —for instance, object pick-and-place problems—, augmented reality, and industrial quality inspection. Despite recent advances in deep learning-based methods, achieving reliable pose predictions remains challenging due to occlusions, clutter, and inherent object symmetries. Probabilistic models have recently been proposed [1] to address some of these challenges by representing uncertainty as distributions over the rotation space SO(3), the special orthogonal group of 3D rotations. However, regardless of the predictive framework employed, a central open question persists: are the predictions based on the correct visual evidence?

While pose estimators can produce high-confidence predictions, such predictions may still be unreliable if they are derived from spurious correlations in the background or irrelevant image regions. This issue is particularly problematic in approaches that train models using synthetic data to solve real-world use cases. This kind of training can bias the model toward synthetic-only cues and cause underperformance in real-world applications; this mismatch is known as the synthetic-real domain gap. Although domain adaptation techniques have been proposed to address this problem, it is also essential to ensure that unexpected anomalies in the object's appearance do not affect the predictions or, if they do, that these predictions can be filtered.

In the probabilistic pose estimation context, confidence metrics, such as likelihoods or entropy of predicted distributions, capture the model's internal uncertainty but provide no direct insight into whether the visual reasoning process is sound. To address this gap, eXplainable AI (XAI) techniques can serve as a powerful diagnostic tool.

In this work, we adapt a gradient-based explanation method to the probabilistic pose estimation setting. By generating saliency maps for the rotation predictions, we can visualize which regions of the input image most strongly influence the network's outputs. This representation enables a fine-grained inspection of whether the model focuses on the target object or instead relies on irrelevant cues, such as background textures or stains. Crucially, these explanations allow us to go beyond uncertainty quantification and introduce an additional filtering stage: pose predictions with unsatisfactory explanation patterns can be systematically identified and discarded.

Thus, the goals of this work are the following: (1) Investigate whether explainability techniques can reliably detect pose prediction errors without requiring ground-truth labels; (2) Propose and evaluate comparison strategies in three spaces (2D image, 3D model, rotation distribution) to filter unreliable predictions; and (3) Analyze the sensitivity of these methods to texture variability.

The remainder of the paper is organized as follows: Section II reviews the state of the art in explainable AI for pose estimation. Section III describes the material and methods employed. Section IV presents and discusses the experimental results. Section V concludes the paper and outlines future work.

II. State of the Art

XAI techniques produce saliency maps highlighting image regions driving a model's decision [2]. Gradient-based methods like Grad-CAM [3] yield coarse localization maps; combined with guided backpropagation [4], Guided Grad-CAM retains fine structural detail. Other methods include Integrated Gradients [5], DeepLIFT [6], and LRP [7]. Attention-based methods [8] leverage internal weights, while Score-CAM [9] uses forward-pass scores. SmoothGrad [10] reduces noise via averaged maps, whereas perturbation approaches like LIME [11] or SHAP [12] are computationally expensive.

Attribution maps help verify if predictions rely on relevant cues. Quantitative metrics such as Over-MAP [13] measure attention–segmentation overlap, and PoseIG [14] uses specialized indices to quantify attribution focus. Filtering unreliable predictions based on these metrics is related to our approach. Similarly, UA-Pose [15] suppresses unreliable pose fits using geometric occlusion. However, saliency maps can sometimes misrepresent true decision drivers [16]–[18]. Extending XAI to 3D pose estimation remains largely unexplored [13]–[15]. Here, we adapt Guided Grad-CAM to a probabilistic 3D-pose

 ISBN: 978-1-68558-395-8

model, hypothesizing that predicted orientations correlate with extracted spatial saliency patterns.

III. Material and Methods

This section explores the datasets employed to measure the quality of our proposed prediction quality scores and the techniques employed. A schematic representation of the pipeline followed is shown in Figure 1.

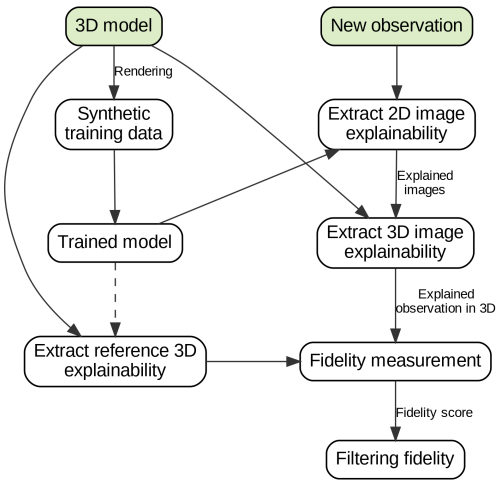


Figure 1. Pipeline of the prediction quality measurement.

A. Datasets

We selected two objects from prior work [1][19] to stress-test two distinct failure regimes that commonly arise in industrial 6D pose estimation: *(i)* a *geometric ambiguity* case and *(ii)* a *domain-gap* case. These regimes were chosen deliberately because they represent the two most prevalent sources of unreliable predictions: ambiguity in object geometry and mismatches between training and deployment domains. Furthermore, the selected objects—a cylinder and a sphere—are geometric primitives commonly encountered in industrial inspection and robotic manipulation scenarios.

Geometric Ambiguity (Cylinder): The cylinder features a square carving on one base and a triangle carving on the other base. This geometric configuration induces multi-modal rotation distributions per view, as different orientations may yield similar visual appearances. For example, 90° rotations around the cylinder's axis can leave the square carving visually unchanged. This regime tests whether explainability techniques can correctly identify when a prediction relies on insufficient geometric information.

Domain Gap (Sphere): The sphere has a "T"-shaped carving, but the real-world texture noise on its surface is absent from the synthetic training data. This regime tests the system's ability to detect when predictions are based on spurious features (texture noise) rather than the intended carving pattern. The noisy texture simulates real-world conditions where training data may not fully capture object appearance variations; the model may confuse backside texture patterns with the "T" carving, producing confident but incorrect 180° rotation predictions.

These objects were selected because they: (1) cover two fundamental failure modes in pose estimation: geometric ambiguity and domain mismatch; (2) allow clear evaluation of explainability techniques via identifiable visual carvings; and (3) have been validated in previous literature [1][19].

While these two objects cover important failure regimes, we acknowledge that they do not exhaustively represent all object types (e.g., texture-only objects without geometric keypoints, or highly articulated objects). Generalization to such categories remains future work.

Models are trained on synthetic [20][21] and CycleGAN domain-adapted renders [19]; real captures (Figure 2) are used for evaluation only.

B. Pose Estimation Model

The probabilistic pose estimation model from [1] predicts rotation probability distributions over the discretized SO(3) space. Translation is out of scope; it can be approximated from multicamera geometry.

C. 2D Explanation

Guided Grad-CAM is applied to interpret the Convolutional Neural Network (CNN)'s decision: Grad-CAM gradients of the rotation score with respect to the last convolutional layer localise relevant regions, and guided backpropagation refines them to edge-level detail. Since the rotation classes are not mutually exclusive, the technique is applied to each predicted rotation mode independently. Results for both objects are shown in Figures 3 and 4.

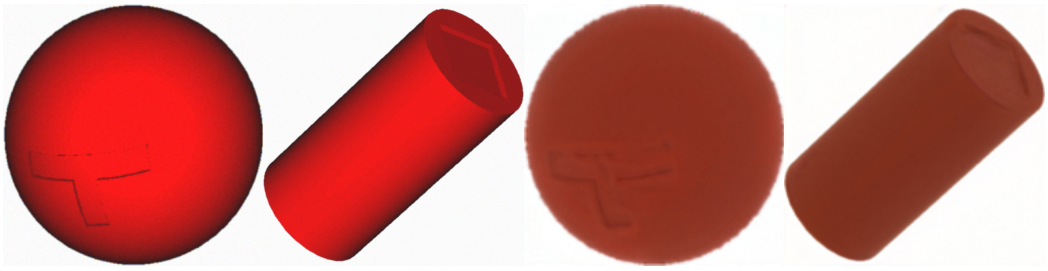
We chose Guided Grad-CAM and Guided Backpropagation over alternative methods for several practical reasons. Gradient-free methods, such as Score-CAM [9], require multiple forward passes per activation map channel, making them prohibitively expensive in our multi-view setup (4 cameras $\times$ multiple rotation modes per image). LRP [7] requires architecture-specific decomposition rules that are not readily available for the graph neural network components of our probabilistic model. Attention-based methods [8] assume transformer-like architectures with explicit attention weights, which our CNN-based backbone does not provide. SmoothGrad [10] could complement our gradient-based approach and is considered for future investigation, as it may reduce the noise observed in our saliency maps (see Section V).

D. 3D Explanation

To extract the 3D explanation, the explanation process computes explainability at the pixel level using the 2D explanation method described above. Then, it uses the network's pose prediction to project the 2D explanations to the object's reference 3D model, which was used to train the pose prediction model. Due to the object's geometry and the camera system setup, all 3D points are visible to at least one camera. Many of them are captured by more than two cameras, so we can use several cameras to assign relevance values.

Gradient-based explainability methods often produce noisy activations, marking pixels as relevant even when they do not truly influence the prediction. To minimize the impact of a single camera's noise on the explained model, we assign the minimum value across all cameras. This approach implies that all cameras seeing a 3D point should agree that it is relevant.

Once a 3D model contains a relevance value per 3D point, the process compares the value obtained per 3D point against a reference explained 3D model.



(a) Synthetic capture of "T"-carved sphere (b) Synthetic capture of the carved cylinder (c) Realistic-transformed capture of "T"-carved sphere (d) Realistic-transformed capture of the carved cylinder

(e) Real capture of "T"-carved sphere (f) Real capture of sphere's noisy backside texture (g) Real capture of cylinder's triangle carving (h) Real capture of cylinder's square carving

Figure 2. Dataset samples showing synthetic, domain-adapted, and real captures for both objects.

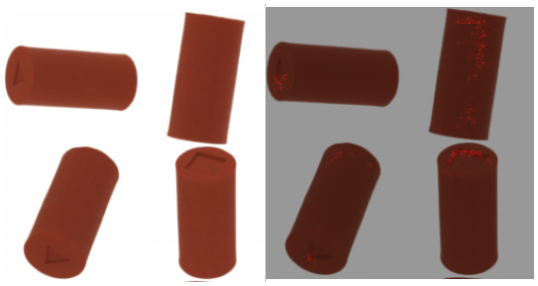


Figure 3. Visualization of Guided GradCAM activations over the cylinder.

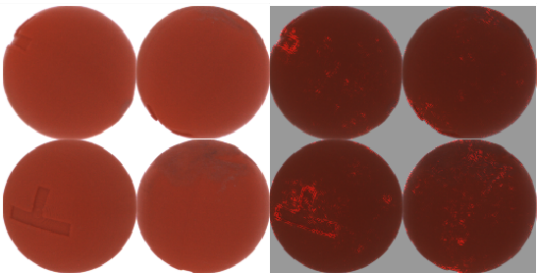


Figure 4. Visualization of Guided Backprop activations over the sphere.

1) Extracting the Reference Explainability: The reference model marks as relevant every 3D point belonging to a different orthogonal triangle of the training mesh, avoiding the noise introduced by averaging multiple explained synthetic batches.

2) Comparing Against Reference Explained 3D Model: At inference, an explained 3D model is extracted using the predicted (not ground-truth) pose; pixel–3D-point misalignment encodes the pose error signal. Three metrics are evaluated: Pearson correlation, dot product, and IoO (see Section III-E).

E. Theoretical Framework of the Explanation Filtering

The theoretical framework for filtering incorrect pose predictions using explainability compares an observed explanation with an ideal reference. We define the following notation:

- I: Ideal 3D explanation (reference model).
- O: Observed 3D explanation (from inference).
- ΔI: Noise in the ideal explanation generation process.
- ΔO: Noise in the observed explanation generation process.
- ΔR: Divergence in explanation due to rotation prediction error.

The similarity score S between reference and observation is computed as:

$$S = f(I + \Delta I, O + \Delta O + \Delta R) \tag{1}$$

where f is an evaluation metric (e.g., Pearson correlation, dot product, or IoO). The Intersection over Observation (IoO) metric quantifies how much of the observed explanation aligns with the reference. Unlike Intersection over Union (IoU), IoO does not penalize missing activations in the observation, focusing instead on whether observed activations fall within expected regions. This feature is advantageous because gradient-based explanations may under-activate some relevant regions while still being correct. The IoO metric is defined as:

$$IoO(I, O) = \frac{|I \cap O|}{|O|} \tag{2}$$

where $|I \cap O|$ represents the set of 3D points activated in both the reference and observation, and $|O|$ is the total activation in the observation.

IoO is well-suited to penalize False Positives (FPs) as background pixels that are activated but do not activate in the reference reduce the metric score. However, IoO does not penalize missing activations. While this is advantageous for incomplete pixel activations —having all reference points matched is unnecessary in some cases and the model is usually capable of adequately predicting using a subset of the relevant pixels— it can mask cases where the model fails to attend to critical keypoints altogether. For symmetric objects, the model may attend to only one symmetric feature (e.g., only the square

carving of the cylinder), yielding a high IoO while missing the discriminative triangle carving that would disambiguate rotations.

To evaluate this trade-off, we evaluate all predictions using three complementary metrics across three comparison spaces. This strategy helps identify cases where a single metric might be misleading.

Our goal is to threshold S to detect large values of ΔR. The key assumption is that ΔR correlates with the rotation prediction error: a larger error produces larger ΔR, because the misaligned pose causes the 2D explanation to project onto incorrect 3D surface regions. However, this correlation can weaken: (i) For the cylinder, for instance, rotations of 90° or 180° around its axis may leave one carving unchanged, producing similar explainability patterns despite non-zero error if the other carving is not attended to. (ii) For the sphere, backside texture noise may produce noisy activations that might be confused with the "T" carving, causing low ΔR at approximately 180° error.

The error progression experiments (Tables I and II) synthetically verify this correlation by perturbing pose predictions.

The following properties hold: (1) $\Delta I < \Delta O$, as ideal explanations can average over multiple synthetic batches; (2) $\Delta I = 0$ is achievable via manual annotation; (3) exact alignment implies $\Delta R = 0$; (4) if $O \subseteq I \Rightarrow f(I, O) = 1$; and (5) $O \cap I = \emptyset \Rightarrow f(I, O) = 0$.

In practice, we can perform this comparison either in the image space or by projecting the image attributions to the training reference model. We measured the results for both approaches. In 3D space, we measured the Pearson correlation between the extracted 3D reference explanation and the observed explanation projected to the 3D model. In 2D image space, we render the 3D reference explanation using the predicted pose, and compute the Pearson correlation between the raw observed image attributions and the rendered reference explanation.

F. Filtering Predictions Using a Distribution Comparison with the Reference

Another filtering strategy compares probability distributions: (1) infer the observation's rotation/translation, (2) render the reference model in the predicted pose, (3) infer the rendered images' distribution, and (4) measure difference. We use two metrics: thresholded blob matching and Pearson correlation.

The approach of comparing distributions using thresholding comprises the following steps. First, the distributions are thresholded utilizing a fraction of the distribution's maximum confidence (between a fifth and a fifteenth), seeking to maximize the number of remaining blobs. Then, we reduce the mask to the likeliest rotations for each thresholded distribution's blob. We then dilate the points around the most likely rotations before comparing the overlap between each mask in the reference and observation distributions. The metric is the ratio of the intersection of blobs in the reference and observation distributions to the total number of blobs. A sample of the thresholding process is shown in Figure 5.

Regarding the second metric, the Pearson correlation, it does not require thresholding; thus, it is applied directly to compare the reference and observation distributions. Therefore, two rotation distributions, such as the ones shown in Figure 5(b), are directly compared using Pearson correlation. The distributions are compared in their original SO3 space, not in the image space used to represent them in Figure 5.

Figure 5 visualizes rotation distributions as unwrapped 3D unitary spheres: red is X-axis, green is Y-axis, intensity is likelihood. Views of the square yield 4 Y-axis solutions; views of the triangle yield 3. Views without features assign equal likelihood at 90 degrees.

IV. RESULTS

The goal of our evaluation is to determine whether the proposed metrics can separate predictions with low rotation error from those with high rotation error, without access to ground-truth labels. We quantify this separation using the Area Under the Receiver Operating Characteristic Curve (AUROC). A prediction is labelled *positive* (valid) when its rotation error is below 10° and *negative* (invalid) otherwise. An AUROC of 1.0 indicates perfect separation—i.e., the metric can distinguish all correct predictions from incorrect ones—while 0.5 corresponds to random guessing.

We evaluate two complementary experimental setups. First, real-world test predictions use actual sensor captures with ground-truth annotations to assess performance under realistic conditions. Second, error progression experiments synthetically perturb ground-truth poses with increasing rotation offsets (from 0° to 180°) and measure the metric response. These controlled perturbations ensure coverage of the full rotation-error spectrum and allow verifying the monotonic relationship between rotation error and metric degradation, independently of the model's natural error distribution.

In the following figures, we refer to the comparison scores described in Sections III-E and III-F as the "Similarity Score." Each scatter plot shows individual predictions, with the x-axis representing rotation error and the y-axis representing the similarity score. An effective filtering metric should show a clear downward trend: high similarity scores for low rotation errors and low scores for high errors.

1) Cylinder: This section presents the cylinder object results. Table I summarises the AUROC values for each comparison method.

TABLE I. CYLINDER AUROC COMPARISONS.

Space	Comparison	Real AUROC	Syn. Prog. AUROC
3D model	Pearson	0.571	-
3D model	Dot product	**0.968**	-
3D model	IoO	**0.968**	0.965
Image	Pearson	0.524	-
Image	IoO	**0.937**	0.970
Dist.	Pearson	0.825	**1.**
Dist.	Blob matching	**0.873**	-

Filtering Using Fidelity: We will begin evaluating the explainability results for the cylinder. Figure 6 shows the quality of fidelity metrics to separate between low and high rotation losses. Regarding AUROC, both 3D-space and image-space metrics achieve large AUROC values. In both cases, the Pearson comparison technique achieved the worst results, with the dot product and IoO being the best comparison techniques for the cylinder, achieving 0.968 AUROC for both the dot product and IoO in 3D space and 0.937 for IoO in image space.

It should be noted that only 3 evaluated real-world samples exceeded 10° error, limiting statistical reliability. However, high

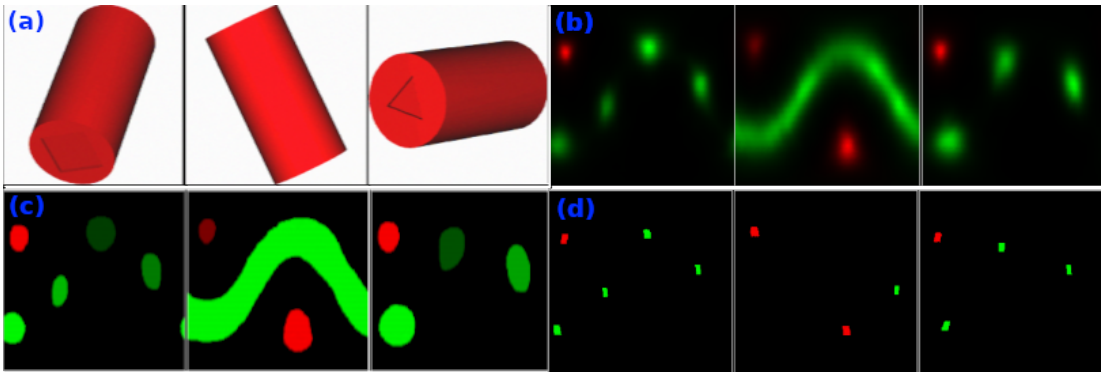


Figure 5. Process of filtering predictions using a distribution comparison. (a) Multi-camera object captures. (b) Rotation distributions. (c) Thresholded distribution blobs. (d) Maximum dilated blobs.

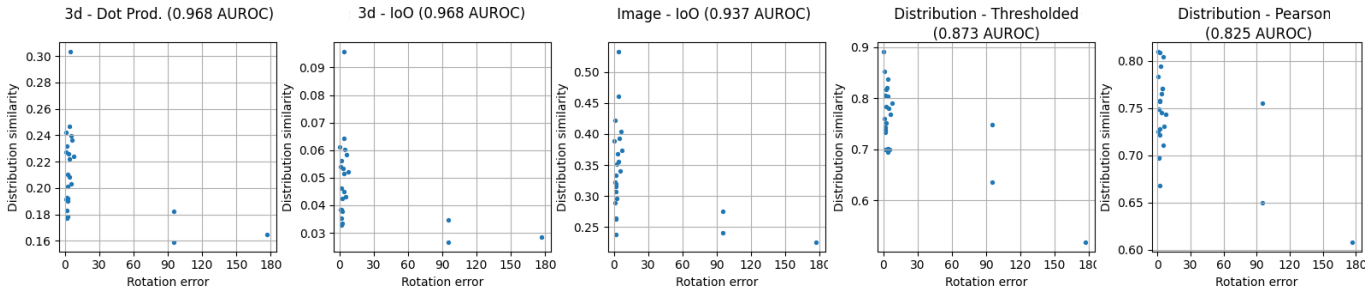


Figure 6. Some of the fidelity and distribution-space metric scores for the cylinder object's model.

values (0.968) indicate strong separation. Error progression experiments confirm monotonic relationships between error and metric score.

Filtering Using Reference Predictions: This section measures the quality of filtering predictions by comparing the observation's probability distribution estimate with the corresponding estimate from the reference model, as explained in Section III-F.

As shown in Figure 6, filtering using the reference predictions would outperform methods employing fidelity metrics, were it not for an observation at 90 degrees that achieved a higher metric score. Both comparison approaches, i.e., thresholding and Pearson, achieve similar results (0.873 and 0.825 AUROC, respectively).

2) Sphere: This section presents sphere object results for the challenging domain-gap scenario. Table II summarises the AUROC values.

TABLE II. SPHERE AUROC COMPARISONS.

Space	Comparison	Real AUROC	Syn. Prog. AUROC
3D model	Dot product	0.788	0.788
3D model	IoO	0.776	-
Image	Pearson	**0.846**	0.856
Image	IoO	0.814	-
Dist.	Pearson	0.788	**0.990**
Dist.	Blob matching	**0.849**	-

Filtering Using Fidelity: Guided Backprop provides the best activations for the sphere. As shown in Figure 4, noisy patterns in the real samples are inappropriately considered relevant because they were unseen during training.

Results for the fidelity metric tracking — both in 3D and image spaces — can be seen in Figure 7. Rotation losses are more distributed than in the cylinder's case, as the object exhibits more texture noise and does not have evenly

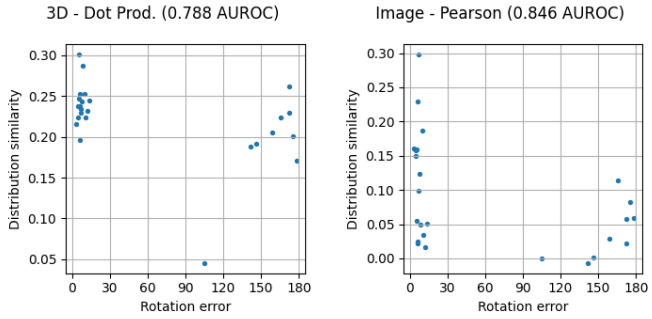


Figure 7. Some of the fidelity metric scores for the sphere object's model.

spaced possible solutions. Rotation losses near 180° represent predictions where the model confused the noisy backside texture with the "T" carving: the explanation activates the backside texture region, which spatially aligns with where the "T" carving would be if the object were rotated 180°, producing a misleadingly high similarity score.

The results over the sphere are less separable than in the cylinder's case, reflecting the additional challenge posed by domain-gap noise. The 3D-space fidelity metrics achieved similar AUROC values (0.788 for dot product and 0.776 for IoO), both lower than for the cylinder. This degradation occurs because projecting noisy 2D explanations to the 3D model amplifies spurious activations through the min-aggregation process (Section III.C). Image-space metrics outperformed 3D-space ones, with Pearson correlation achieving the highest fidelity-based AUROC (0.846). This reversal compared to the cylinder indicates that when texture noise dominates the explanation, direct image-space comparison avoids the projection-induced artifacts that degrade 3D metrics. The noisy texture provokes the spike in metric scores at approximately

 ISBN: 978-1-68558-395-8

160° rotation loss, as visualized in Figure 4.

Filtering Using Reference Predictions. Employing predictions instead of fidelity to filter predictions yields better results, as an AUROC of 0.849 can be achieved using blob matching (Figure 8), outperforming the fidelity-based metrics but only by a small margin. Although it does not achieve a perfect separation of low and high rotation loss predictions, the decreasing trend is more perceptible than in the fidelity case.

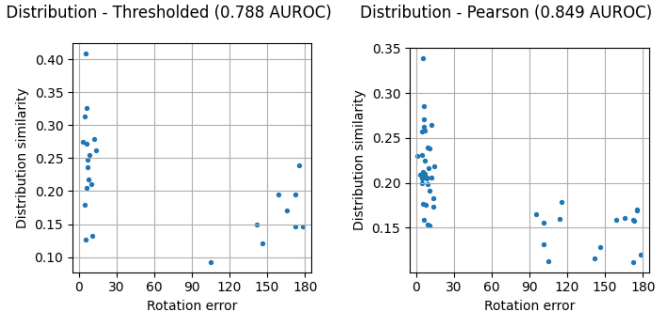


Figure 8. Distribution-space metrics for the sphere object's model.

A. Discussion and Method Selection

The experimental results reveal distinct performance patterns across the two object types and three evaluation spaces. For the cylinder (geometric ambiguity regime), 3D-space metrics (IoO and dot product) achieve the highest AUROC values (0.968). The strong performance stems from the deterministic projection of keypoints. Image-space metrics achieve similar results (0.937) but avoid intermediate 3D projection artifacts. Distribution-space metrics achieve perfect monotonic separation in synthetic experiments.

Conversely, for the sphere (domain-gap regime), 3D-space metrics perform significantly worse (0.776–0.788) because projection amplifies texture noise. Here, image-space Pearson correlation outperforms 3D metrics (0.846), indicating robustness to domain-gap artifacts. Furthermore, distribution comparison (blob matching) achieves 0.849 AUROC, outperforming all fidelity methods by bypassing the explanation pipeline entirely to directly compare prediction distributions.

Method Selection Recommendation: We recommend a two-stage filtering approach. First, for objects with clear geometric keypoints, use 3D-space fidelity metrics (IoO or dot product) as the primary filter. Second, for objects with texture variability or domain gaps, prioritize distribution-space comparison (blob matching) or image-space metrics (Pearson correlation) to handle explanation noise.

V. Conclusion and Future Work

We presented a framework for filtering pose estimation predictions by leveraging XAI and distribution comparison without ground-truth labels. For the cylinder, 3D projected fidelity metrics demonstrated reliable error detection from saliency alignment. For the sphere, gradient-based attributions were dominated by domain-shifted texture noise, making distribution comparison more robust.

Together, these results suggest a two-stage reliability pipeline: first validate whether attributions are stable and object-centred, then apply the appropriate filter. Future work should explore: (i) extracting the 3D reference from validation data, enabling texture-keypoint objects; (ii) improving attribution precision through alternative methods like SmoothGrad [10], Score-CAM [9], LRP [7], and attention mechanisms [8]; and (iii) extending evaluation to other objects.

Acknowledgment. This work has been carried out within the framework of project GUARDIANES with grant number CER-20251017, funded by the *Centro para el Desarrollo Tecnológico Industrial* (CDTI).

References

[1] O. Del-Tejo-Catala *et al.*, "Probabilistic pose estimation from multiple hypotheses", *IEEE Access*, vol. 11, no. April, pp. 64 507–64 517, 2023.

[2] T. I. Amosa *et al.*, "Multi-camera multi-object tracking: A review of current trends and future advances", *Neurocomputing*, vol. 552, p. 126 558, 2023.

[3] R. R. Selvaraju *et al.*, "Grad-cam: Visual explanations from deep networks via gradient-based localization", in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 618–626.

[4] T. J. Springenberg *et al.*, "Striving for simplicity: The all convolutional net", *CoRR*, vol. abs/1412.6806, 2014.

[5] M. Sundararajan *et al.*, "Axiomatic Attribution for Deep Networks", in *Proceedings of the 34th International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, vol. 70, PMLR, 2017, pp. 3319–3328.

[6] A. Shrikumar *et al.*, "Learning Important Features Through Propagating Activation Differences", in *Proceedings of the 34th International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, vol. 70, PMLR, 2017, pp. 3145–3153.

[7] A. Binder *et al.*, "Layer-wise relevance propagation for neural networks with local renormalization layers", Apr. 2016.

[8] H. Zhang *et al.*, "Diverse Attention for Explanations and Robustness", in *International Conference on Learning Representations (ICLR)*, 2021.

[9] M. Chen *et al.*, "Score-CAM: Score-Weighted Visual Explanations for Convolutional Neural Networks", in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[10] D. Smilkov *et al.*, "SmoothGrad: removing noise by adding noise", *arXiv preprint arXiv:1706.03725*, 2017.

[11] M. Ribeiro *et al.*, ""why should i trust you?": Explaining the predictions of any classifier", in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1135–1144.

[12] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions", *arXiv preprint arXiv:1705.07874*, 2017.

[13] C. Kantor *et al.*, "Over-map: Structural attention mechanism and automated semantic segmentation ensembled for uncertainty prediction", *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 17, pp. 15 316–15 322, May 2021.

[14] Q. He *et al.*, "Analyzing and diagnosing pose estimation with attributions", in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 4821–4830.

[15] M.-F. Li *et al.*, "Ua-pose: Uncertainty-aware 6d object pose estimation and online object completion with partial references", in *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025, pp. 1180–1189.

[16] J. Adebayo *et al.*, "Sanity checks for saliency maps", *Advances in Neural Information Processing Systems*, vol. 31, no. NeurIPS, pp. 9505–9515, 2018.

[17] P. J. Kindermans *et al.*, "The (un)reliability of saliency methods", *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11700, pp. 267–280, 2019.

[18] A. Ghorbani *et al.*, "Interpretation of neural networks is fragile", *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 1, pp. 3681–3688, 2019.

[19] O. Del-Tejo-Catala *et al.*, "Synthetic-real domain adaptation for probabilistic pose estimation", *Computer Science Research Notes*, vol. 31, no. 1-2, pp. 127–136, 2023.

[20] Y. Xiang *et al.*, *PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes*, Nov. 2018.

[21] W. Kehl *et al.*, "SSD-6D: Making RGB-Based 3D Detection and 6D Pose Estimation Great Again", in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-October, Nov. 2017, pp. 1530–1538.

Building Confidence: An Ontological Approach to Assurance of Safety-Critical Systems

Odd Ivar Haugen
Group Research and Development department, DNV AS
Trondheim, NORWAY
e-mail: odd.ivar.haugen@dnv.com

***Abstract*—This paper presents an ontological framework for the assurance of safety-critical systems, focusing on the foundational relationship between knowledge, confidence, and risk. Society is growing increasingly intolerant of risk in high-tech systems; therefore, stakeholders must be provided with justified confidence in a system's safety. This confidence is not based on mere compliance with standards and guidelines, but on a robust assurance framework that demonstrates that the system behaves safely. The proposed ontology meets this need by defining assurance as a process that generates explicit knowledge to reduce uncertainty. At this framework's core, knowledge acts as the "hub" of assurance; when systematically represented in an assurance case, it directly influences stakeholder confidence. In this assurance, we generate knowledge whose objectivity provides a robust metric for justifying claims. This approach ensures that arguments supporting system safety are not only coherent but also demonstrably strong, linking the assurance effort directly to risk levels. This ontological model provides a comprehensive and systematic methodology for demonstrating safety by connecting the elicitation of system requirements to the justification of claims. This work, therefore, offers a structured path for building and communicating grounds for justified confidence in the responsible deployment of complex and novel systems.**

Keywords-assurance; confidence; knowledge; risk; safety.

I. INTRODUCTION

High-tech systems, with their increasing complexity and societal integration, necessitate rigorous methods to ensure their safe and responsible operation. In safety-critical systems, where failures can have catastrophic consequences, stakeholders like operators, regulators, and the public must have justified confidence that the system will behave as intended. However, traditional approaches that focus on compliance with established standards often fail to address the novel risks and emergent behaviours of modern technologies. This failure creates a need for a more foundational approach to safety assurance.

The field lacks a clear, underlying framework that explicitly defines how assurance activities build this necessary stakeholder confidence. While common practices like developing safety cases exist, they can become procedural exercises if they lack a robust ontology that connects arguments and evidence to a tangible reduction in uncertainty and risk. To make assurance efforts both efficient and effective, this paper addresses the need for a systematic model that explains the core relationships between knowledge, risk, uncertainty, and confidence.

Existing assurance methodologies provide structures for arguing about safety, but they do not always articulate the epistemic principles that govern why these arguments should be considered trustworthy. The core limitation of current practices is the frequent disconnect between the assurance artefacts produced and the fundamental goal of cultivating a justified belief in the system's safety among diverse stakeholders. The need to bridge this gap motivates our work, which establishes a clear line of reasoning from stakeholder concerns about potential losses to the justified claims made about system behaviour.

This paper introduces a comprehensive ontological framework for the assurance of safety-critical systems, positing that assurance is fundamentally an epistemic activity. This framework generates explicit knowledge to reduce uncertainty about a system's properties. Our central thesis is that knowledge serves as the "hub" of assurance; when systematically gathered, analysed, and presented, this knowledge provides the robust and justifiable grounds for stakeholder confidence.

To develop this framework, we first model the intrinsic connections between risk, confidence, and uncertainty, demonstrating how generating knowledge directly reduces epistemic uncertainty. We advocate for a systems approach, utilising the CESM metamodel (Composition, Environment, Structure, Mechanism) to analyse emergent properties, such as safety. As a core contribution, this work establishes objectivity as a multi-dimensional metric for evaluating the strength of knowledge. Finally, we organise this knowledge within a structured assurance case. This assurance case systematically links claims about system safety to their substantiating arguments, thereby providing a scrutable and justified basis for confidence.

This paper is structured as follows. Section II introduces the main concepts of assurance. Section III provides an overview of assurance and confidence. Section IV discusses the relationship between assurance and risk. Section V presents the systems approach and the CESM metamodel. Section VI addresses epistemology and justification. Section VII introduces objectivity as a metric of knowledge strength. Section VIII discusses assurance cases. Section IX covers stakeholder objectives and system requirements. Section X concludes the paper and outlines future work.

II. MAIN CONCEPTS OF ASSURANCE

Assurance is about becoming confident that the system behaves in a way that is acceptable to the stakeholders. Here, stakeholders are seen as any person, group of persons, governmental regulator, society, or even the natural environment. In short, it is an entity that is affected by the behaviour of the system.

A claim is a property of interest about the system. The claims can be thought of as system requirements; that is, ”this” is how the system should behave in order for the system to be accepted by the stakeholders. Analysing the previous statement reveals, as a first approach, the four principal criteria that must be in place to achieve acceptance:

1) the system requirements must reflect the interest of the stakeholders,
2) refining these requirements into technical specifications must maintain the essence of these requirements,
3) the system's adherence to these requirements must be secured and adequately substantiated,
4) 1, 2, and 3 must be communicated to the stakeholders or their representatives in such a way that they can make intelligible decisions.

It is clear from the above items that the key to system acceptance is *knowledge*. Indeed, knowledge may be said to be the ”hub” of assurance. The stakeholders must know that the system behaves acceptably. Knowledge is a prerequisite for confidence, which reduces the uncertainty about the system.

Confidence is different from trust. Confidence is something that can be merited through demonstrating adequate capability; trust, however, has to be earned through time; that is, trust is closely connected to an agent's intention. This means that confidence can be merited through demonstrating adequate capability (technical system and responsible agent); trust must be earned through time by a responsible agent adhering to sound and recognised ethical principles.

As assurance is about providing grounds for justified confidence, this paper will therefore focus on how to demonstrate adequate system capability so that the stakeholders can make intelligible decisions based on their knowledge and, thereby, their level of confidence in the system.

It should be noted that assurance is an epistemic activity, while risk management encapsulates both epistemology and intervention in the real world [1] [2].

The system capability, in this context, is equivalent to how the system behaves under normal operation and in abnormal situations.

The system risk is defined as the ”effect of uncertainty on objectives” [3] and reflects the consequences and uncertainties that the system causes losses for stakeholders. The uncertainty is here divided into two types: epistemic and aleatory [4].

Item three in the above list requires that the system behaviour adherence to the requirements is substantiated; that is, claims about the system must be substantiated through sound and relevant argumentation. For an argument to be sound, it must be generated in accordance with acknowledged methodologies using reliable tools and adequately skilled people.

To assess the soundness and the strength of arguments, an assessor not only needs to be a subject matter expert but also needs guidance about what can be regarded as acceptable methods and processes to develop arguments; that is, he needs guidance about the argument's *objectiveness*. A higher degree of objectivity increases the strength of the argument, which is necessary when the risk is high, such as for safety-critical systems.

III. Assurance and confidence - an overview

Confidence can be thought of, in statistical terms, as a quantitative measurement of uncertainty, e.g., an interval indicating the confidence that the value of a parameter is likely to fall within. However, confidence may also be thought of as a feeling that reflects the coherence of the information and the cognitive ease of processing it [5]. Assurance is defined as ”grounds for justified confidence that a claim has been or will be achieved” [6]. The definition does not limit assurance to either type of confidence; hence, assurance addresses both.

Both types of uncertainties pose challenges. The frequentist approach to quantifying uncertainty requires robust statistical data. Here lie a few major obstacles, some of which are: the inherent complexity of many safety-critical systems, the novelty of the technology, statistically significant data from rare events, and assigning probabilities to inherently social aspects.

The second type of confidence also poses challenges. We cannot base decisions concerning the safety and well-being of stakeholders and society on pure feelings but on strong knowledge based on facts and trustworthy evidence.

Therefore, assurance may provide grounds for justified confidence through uncertainty quantification only if based on robust statistics, that is, knowledge about properties of the statistical distribution of the parameter in question, and/or judgemental assessments only if based on sound argument substantiating the truthfulness of the claim.

Therefore, the immediate goal, or primary effect of assurance, is to generate knowledge, knowledge to decrease or establish the uncertainty about a claim, addressing both types of uncertainty when appropriate. A Functional Analysis System Technique (FAST) diagram illustrates the relation between assurance, knowledge and confidence (Figure 1).

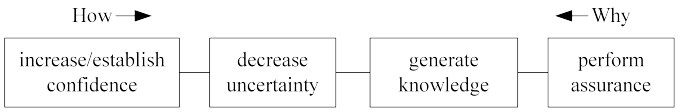


Figure 1. FAST diagram connecting knowledge to confidence.

A FAST diagram is read either way from left to right by asking, *How* is this function achieved? Or, from right to left by asking *Why* does this function need to be achieved.

As knowledge is the ”hub” of assurance, knowledge must be treated systematically and expressed explicitly to enable it to be rigorously scrutinised. This is to avoid that confidence being based on unsubstantiated feelings and pure guesswork. The assurance case is a systematic and explicit way of representing and treating knowledge.

As safety is an emergent property [7], the knowledge about the truthfulness of the claim must address all system aspects that affect emergence. Elements necessary in analysing emergent behaviour in engineered socio-technical systems are encapsulated in the systems approach.

Figure 2 depicts how the different items of assurance are related.

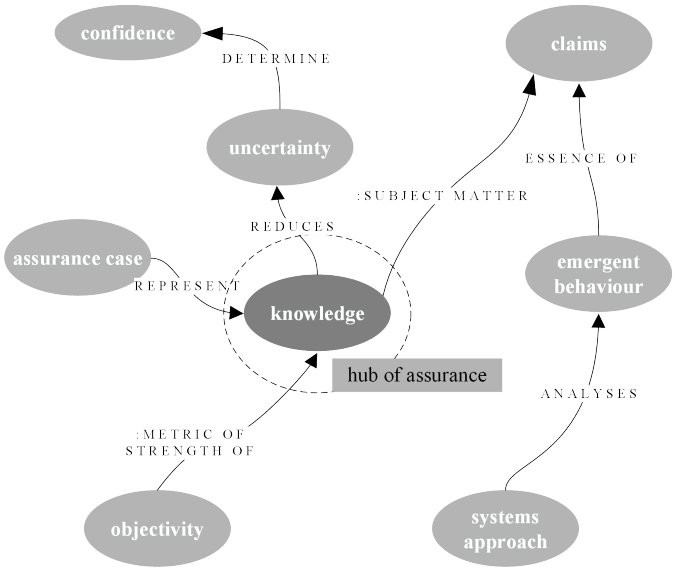


Figure 2. Overview of the ontology of assurance.

Intuitively, the higher the risk that the system poses to stakeholders, the higher confidence we need that it will indeed behave as expected. As knowledge reduces uncertainty and increases confidence, we need a way to assess its strength. Assessing the strength of knowledge is key to adjusting the assurance effort to risk level.

IV. Assurance and System Risk

Figures 1 and 2 showed how knowledge generated in the assurance effort reduces uncertainty, and that uncertainty determines confidence. Moreover, as earlier established, uncertainty is one part of the risk concept. Hence, assurance and risk are connected through uncertainty (Figure 3).

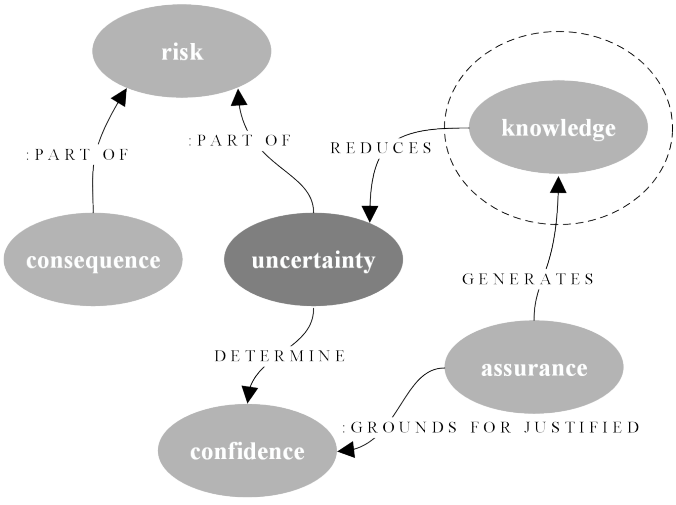


Figure 3. Assurance is connected to risk through uncertainty.

There is, however, another connection in addition to the one mentioned above. In the top right corner of Figure 2, it is indicated that the subject matter of the knowledge is the claim. Claims are statements about system properties that address the system requirements elicited by stakeholders and their concerns and objectives (Figure 4).

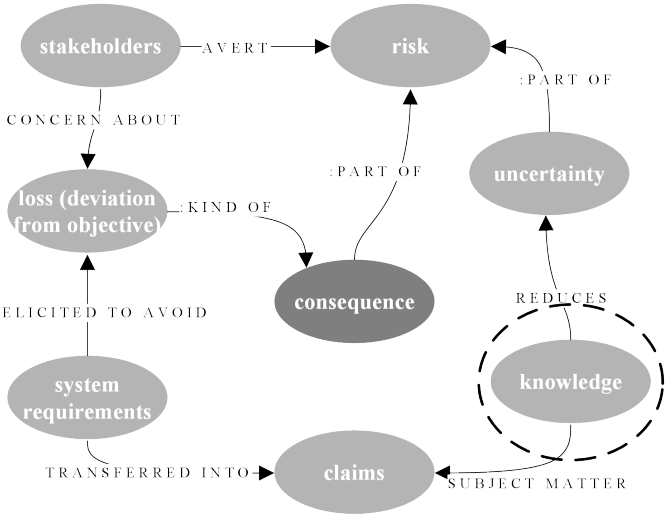


Figure 4. Assurance is connected to risk through claims.

Stakeholders are generally risk avert [5] and are concerned about the consequences of losses. They need adequate confidence that potential losses are acceptable. Assurance addresses these concerns by generating knowledge about the truthfulness of claims made about the system properties.

Risk can be reduced by altering the system design or operational conditions. These risk-reducing strategies affect the consequence and/or the aleatory uncertainty. However, as this paper is concerned with assurance, which is an epistemic endeavour, these two strategies are not further discussed. Their relationship to assurance is discussed in [1]; on the relationship between assurance and risk management.

V. Assurance and the Systems Appoach

A way to understand and analyse complex systems and emergence, is to model the system behaviour in terms of its composition, structure, mechanisms and the environment in which it operates. These system aspects are termed the CESM metamodel [8]:

- **Composition** (C): Collection of all the parts or objects in the system.
- **Environment** (E): Systems outside (excluded from) the target system, but act upon, or are acted upon by, the target system.
- **Structure** (S): The relationships and bonds among the system agents and between the system agents and the environment.
- **Mechanisms** (M): The processes that make the system behave in the way that it does.

The emergent behaviour becomes a function of the above elements; that is, any system s may be modelled, at any given instance, as the quadruple: $\mu(s) =< C(s), E(s), S(s), M(s) >$. As $\mu(s)$ is an emergent property, and emergent properties exist on different levels of abstraction (LoA) [9], the CESM must also be instantiated at these LoAs.

This can be visualised by the system triangle (Figure 5) where the corner of the triangle illustrates "CSM" encapsulated by "E". The "system" in the middle represents $\mu(s)$. $\mu(s)$ emerges, therefore, as a result of the conceptual interaction between the corners of the triangle, but also between the triangle and the environment (E). To move the analysis between the LoAs, a rule-based gradient is used, termed the gradient of abstraction (GoA).

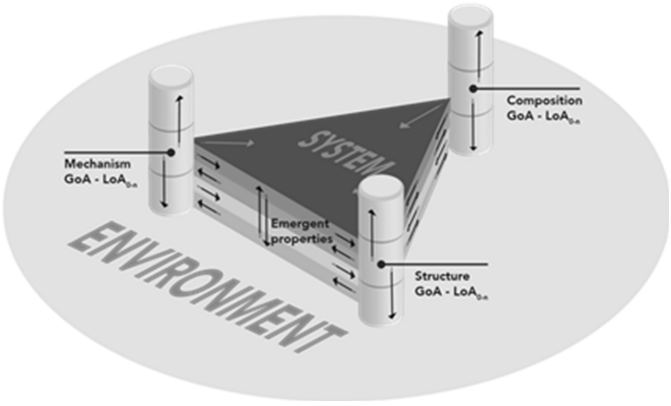


Figure 5. The CESM triangle showing.

For each element in the CESM metamodel, we can assign different system model categories [10]:

- Composition: **Object model** representing the system elements and components and their ontological relationship to each other.
- Environment: Also modelled as a system containing all aspects of the CESM metamodel, which means that the environment must be represented by models representing the composition, structure and mechanisms (our target system is part of the environment of its environment).
- Structure: **Agent model** includes entities, such as controllers, actuators, sensors, humans, and subsystems. The agent concept includes authority, responsibility, goals, concerns, motivation, and wishes (humans).
- Mechanisms: **Function model** represents the operations that must be performed (by the agents) to achieve goals.

Examples of system model instantiation of the agent model is the control structure known from Systems-Theoretic Process Analysis (STPA) [7]. Another agent model may focus more on the agent's goals, motivation, concerns and wishes, like a model used in a stakeholder analysis where social and business aspects are emphasised.

A function model may focus on the preconditions, resources, and timing for achieving it, like the model used in the Functional Resonance Analysis Method (FRAM) [11].

The functional dependencies between functions, like in FAST [12] may be used as GoA to move the analysis between abstraction levels, that is, to represent the system at different LoAs [13].

The systems approach described above, used in assurance, can be summarised by the following statements [2]:

- The conceptual interaction between the system composition (C), environment (E), structure (S), and mechanisms (M) models the system behaviour.
- The kind and number of levels of abstractions (LOAs) used in the modelling is determined by the knowledge sought through the assurance effort.
- The systems approach is used in every aspect of the assurance effort, such as system description, describing the system boundary, describing the environment in which the system is operating, system analysis, verification and validation, and elicitation of system requirements.

Figure 6 depicts the relationship between the systems approach and assurance.

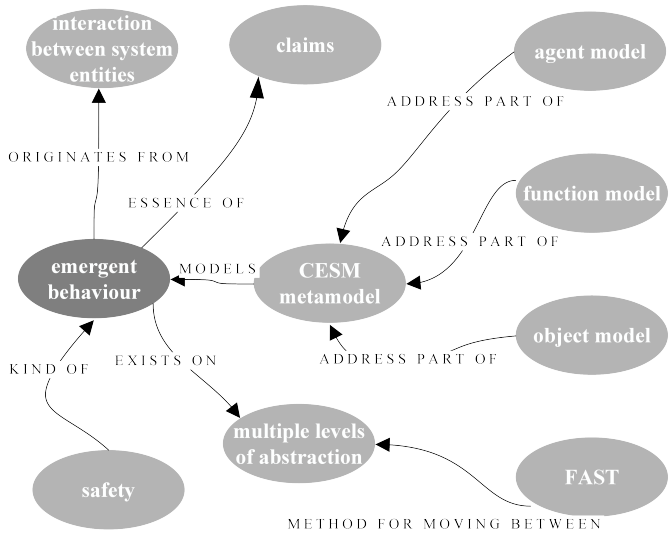


Figure 6. The systems approach is connected to assurance through the claims or requirements.

The system safety requirements are formulated as safety claims. As safety is an emergent property that emerges through the interaction between the system entities, it can be modelled through the CESM metamodel.

VI. Assurance, epistemology and justification

Recall that the concept of risk incorporates, in addition to the consequence, two kinds of uncertainties: epistemic and aleatory. Strengthening the knowledge reduces epistemic uncertainty. If the risk is high, as in safety-critical systems, the argument supporting the claim must be strong. The strength of the argument and, thereby, the strength of the knowledge reduces the epistemic uncertainty and, thereby, the risk.

The classic definition of knowledge is: "Justified True Belief" (JTB). Although this definition has been under scrutiny for centuries and has been shown to have weaknesses [14], it must be linked to accessible facts about the subject matter. Moreover, building confidence through knowledge requires, not only apparently truthful propositions (claims), but also that the reasoning is sound, relevant and adequate; that the proposition is justified: *"Someone who is very confident but for the wrong reasons would also fail to have knowledge"* [15].

The reason for believing that a proposition represents the truth must be justified.

Justification may be thought of as an argument for why we hold certain beliefs or why we think those beliefs are reasonable and true. These justifications may be under the law or before God. However, in the context of assurance, justifying beliefs must be based on knowledge, or, in other words, on epistemic justification [16]. (A safety-critical system needs, of course, to conform to laws and regulations; however, the point is that the justification must be based on knowledge.)

Assurance seeks epistemic justification to establish whether a proposition can be turned into a belief, that is, belief through warranted propositions.

Belief revision is the process of changing beliefs based on new data [17]. It is important to emphasise that good reasoning is no guarantee of truth. Seeking the truth and believing to have found it using sound methods and reasoning is no guarantee of actually having found it.

Justifying a proposition may, in principle, entail an infinite chain of justifications (infinitism): The justification of the justification of the justification... This is, of course, unacceptable. The question, then, is when to stop this chain of justifications.

One strategy is to continue until the supporting justifications become self-evident, that is, propositions that do not need further justification (foundationalism). This kind of justification results in a hierarchy of propositions, and the "bottom" of this hierarchy consists of fundamental propositions, that is, self-justified propositions.

Alternatively, we may ensure that the propositions support each other, that is, the propositions are coherent (coherentism). With this strategy, there are no fundamental propositions. Critiques claim that this strategy can lead to circular argumentation [16].

A reasonable approach is to combine the two strategies, that is, ensuring coherence within the set of propositions and justification, and stopping the chain of justification when reaching a self-justified proposition.

In practice, one may not reach a self-evident fundamental level for several reasons. One reason may be that there is a dispute about whether such a level is actually reached; another reason may be that continuing the chain of justification requires disproportionate resources. Therefore, there may be residual uncertainty as to whether a proposition represents the truth.

Showing compliance towards an international industry standard is often regarded as such a self-justified belief. Providing evidence that a system complies with such a standard is often regarded as adequate for believing a proposition, e.g., that a system is reliable, fair, safe, and secure, as an international standard should reflect good industry practice. However, e.g., artificial intelligence (AI) is a novel technology that, even if there exists a relevant international standard, it may not be regarded as self-justified because the standard itself does not necessarily reflect any industry practice (because there do not exist any such practice), or at least the practice may be inadequate. This means that it might be necessary to continue the justification chain further when assuring novel complex systems, e.g., based on AI.

Other sources of uncertainty include evidence that weakens the proposition or a lack of available evidence. Moreover, other obstacles may hinder the generation of additional evidence, such as technical limitations, ethical concerns, lack of statistical data, or other practical causes.

There is no universal uncertainty threshold for when an agent will accept a proposition and when he rejects it. Moreover, given a justification of a proposition, there is no universal law governing the level of uncertainty an agent will feel about its truthfulness.

Belief revision depends not only on the properties of the justification of the proposition but also on the agent's epistemic state, that is, the agent's required rationality to turn a proposition into a belief, prior belief and any other properties important for the agent to represent facts about the world.

The uncertainty threshold for an agent's belief revision also depends on aspects such as the risk (perceived and/or actual) of accepting or rejecting a proposition (including being indifferent). Moreover, an agent's level of uncertainty, given a justification of a proposition, depends not only on the strength of the justification, but also on aspects such as the degree of being susceptible to cognitive biases [5] and rhetoric. Obviously, we should strive to minimise aspects of belief revision that are unrelated to the properties of the justification.

Perhaps the most commonly known is the so-called confirmation bias, that is, our tendency to seek evidence that confirms our prior beliefs. However, most other cognitive biases are at work, like the illusion of understanding and what you see is all there is (WYSIATI), that is, our tendency of believing that we understand complex topics by filling in the information gaps and the epistemic gaps so that the story becomes compelling and coherent, which leads to confidence in the truthfulness of the story (or proposition in this case).

An agent's prior beliefs cannot, and should not, be controlled and cannot be totally known. Nevertheless, prior belief is central to belief revision. Data-oriented Belief Revision (DBR) [18] (simplified illustration in Figure 7) is a model of belief revision that can illustrate the role of prior belief in belief revision.

After new data is available about a proposition (External data), the data is assessed to determine their relevance and strength, possibly forming a new or updated belief set, termed *belief selection* in Figure 7. This process regulates the interaction between data and beliefs, what to believe in, and with what strength.

As belief revision is tightly connected to the agent's prior beliefs and possible degrees of cognitive biases, we cannot assess the epistemic strength of the justification by appealing to the agent's prior beliefs, or what seems to be "very reasonable" and the like. What seems reasonable is an internal feeling in each agent and is largely based on their current epistemic state.

Instead, the agent needs to be nudged towards sound rationality of assessing uncertainty using a more comprehensive framework of thinking about the level of uncertainty (epistemic strength of the justification), without being forced into an

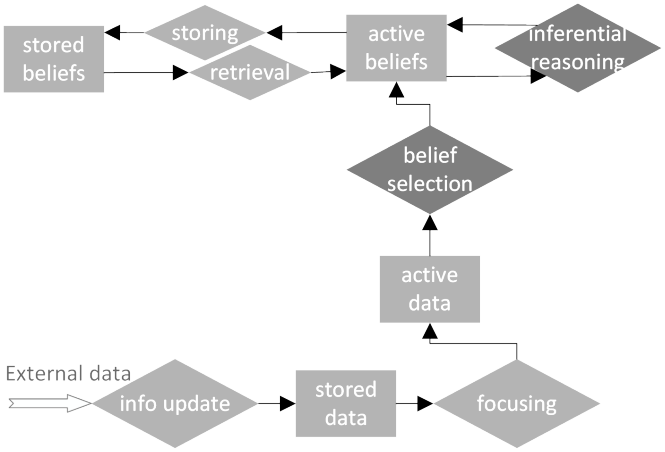


Figure 7. Simplified epistemic processing in DBR [18].

epistemic straitjacket of predefined categories of epistemic levels.

We want to decrease uncertainty as the risk of accepting a false proposition increases. The opposite may not be so obvious, that we also want to decrease uncertainty when risk increases by rejecting a true proposition. Accepting a false proposition, on the one hand, or rejecting a true proposition, on the other, represents assurance risk.

Assurance risk, that is, the risk of making wrong decisions due to weak or inaccurate knowledge, is one kind of risk in the context of assurance. The other kind of risk is the system risk, that is, the undesired consequences with associated uncertainty of operating a system in the real world.

Decreasing uncertainty to the point of accepting a proposition, or in other words, revising one's belief, can be achieved by both strengthening the justification that the proposition is true, and/or by increasing effort in seeking justification that the proposition is false without finding such justification. Sometimes, the only way to justify a proposition p is to find a strong justification that $\neg p$ is not the case.

A famous statement from software testing illustrates this: *Software testing cannot prove the absence of bugs, only their presence*. A proposition that some software code is bug-free p cannot be proven through testing alone. Software testing tries to find bugs, and when no bugs are found, one may start to believe p because one hasn't found evidence that $\neg p$ is the case. However, as most testing is non-exhaustive, not finding bugs does not mean the absence of bugs.

A way to accommodate proper assessment of knowledge built on epistemic justification is through argumentation. While belief revision describes how we should update our beliefs, argumentation is a way to make belief revision occur. "The two concepts are two sides of the same epistemic coin" [19] [18].

VII. Objectivity - a metric of strength of knowledge

By generating knowledge about the system, the epistemic uncertainty about deviation from objective changes, that is, knowledge about how an accident may occur or the potential consequence should it occur. High risk means severe potential consequences combined with a large degree of uncertainty (epistemic and/or aleatory). As knowledge decreases uncertainty, high risk requires strong knowledge, that is, knowledge substantiated with strong grounds for justification.

Justification, and thereby knowledge, is, among other things, based on artefacts representing the system and its properties, together with how these artefacts are interpreted, that is, the reasoning used to conclude based on these artefacts. Artefacts, such as training data, algorithms, source code, and system descriptions, may represent the system directly. Other kinds of artefacts may indirectly represent it, e.g., artefacts generated through verification, such as test cases, test results and results from inspections and reviews. The strength of knowledge is directly linked to these artefacts and the process of generating and collecting them.

Distinguishing weak from strong knowledge requires a metric by which the strength of knowledge can be assessed. By comparing the definitions of knowledge and assurance, we recognise the similarities. Both definitions contain the term "justified": The degree of justification for a true belief (knowledge) - the grounds for justified confidence (assurance). Degree of justification is central in assessing both strength of knowledge and degree of confidence (via uncertainty as shown in Figure 2). A high degree of confidence requires strong ground for justification.

Objectivity encapsulates the aspects important for assessing the degree of justification, that is, the strength of knowledge. Hence, the strength of knowledge is measured through the degree of objectivity. The likelihood that the result of an enquiry represents the truth increases if it is conducted in an objective manner, including the artefacts produced and used in that enquiry.

Ensuring consistency and repeatability in our enquiries requires that the concept of objectivity be described. Objectivity in this context is a multi-dimensional, non-orthogonal and non-binary concept [20]. Hence, objectivity cannot be treated in a reductionist manner.

There are three categories (i.e., dimensions) that lay out the space of objectivity [20] [14] (Figure 8):

1) properties and processes by which the artefacts are generated
2) reasoning, or the thinking about those artefacts
3) social processes concerning items 1 and 2.

Item 1 is about interacting with the system and its stakeholders during its entire lifecycle. It is about the choice of methods, how they are applied, and how those decisions influence the properties of the outcomes, that is, the artefacts. This category also includes procedures, methods, techniques, first principles in physics, standardised equations, algorithms, etc.

Item 2, this category is about how people and organisations think and the reasons and positions they take based on their interests and roles. This includes the involved assurance agent's values and independence from the developer.

Item 3 is about the social processes that advocate different viewpoints, such as agreement among subject-matter experts

on the suitability and correct use of methods for generating artefacts and on how to think about those artefacts. This kind of objectivity can be thought of as a form of inter-subjectivity and is strengthened if the group consists of individuals with different but relevant competencies. The content of standards is a result of such agreements.

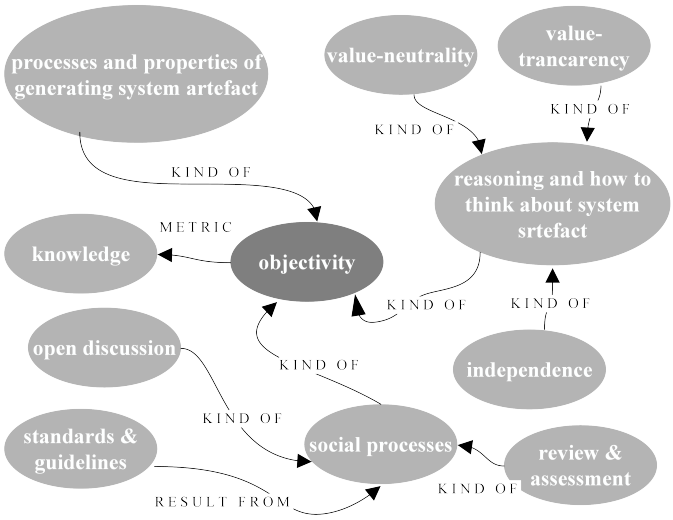


Figure 8. Categories of objectivity.

An important activity in assurance is the generation and collection of evidence through verification and validation (V&V). V&V is described through two properties: 1) The level of intensity in the V&V effort, and 2) the level of rigour in the V&V effort [21]. V&V intensity is connected to the size of the scope, the number of system artefacts investigated, and the level of V&V involvement in each phase of the system lifecycle. V&V rigour is connected to comprehensiveness and thoroughness, leaving less room for logical inconsistencies and contradictions in the results, that is, performed with different levels of formality concerning techniques and documentation. One useful metaphor describing the relationship and difference between the two properties may be that increased V&V intensity makes the mesh width smaller and smaller, while increasing the V&V rigour means that each mesh is investigated closer and closer.

The output from the V&V effort is the evidence representing the system properties of interest, such as safety, reliability, robustness and security. V&V intensity and rigour affect the evidence properties [21] such as quality, capability, and coverage.

Confidence is a result of the assessment of the strength of justification and knowledge through the degree of objectivity. Furthermore, through the V&V intensity and rigour, and the resulting evidence properties. The assessment cannot be a simple checklist, which results in a numerical score aggregated as a simple sum or a single-dimensional category. The strength of knowledge must be assessed in each particular project in the context of a totality. That is, the strength (of knowledge) is not a resultant property of the degree of objectivity (and V&V), but emergent. Assessing the truthfulness (strength of justification and knowledge) of claims made about emergent properties in novel, complex safety-critical systems depends on the judgement of experts in the relevant disciplines. It is guided by the objectivity criteria described here.

This position does not preclude the use of quantitative or probabilistic measures where they are epistemically justified and appropriate; rather, it asserts that no single quantitative score can substitute for expert judgement when assessing the strength of knowledge concerning emergent properties in complex safety-critical systems.

VIII. Assurance case - a systematic way to represent knowledge

The assurance case is a way to represent knowledge (Figure 9 and Figure 2). At its core, an assurance case consists of a hierarchy of claims and arguments, including evidence that substantiates those claims. The claims are equivalent to the before-mentioned propositions, and the argument is equivalent to the before-mentioned justifications. Moreover, claims can be understood as a reformulation of system requirements. A question may be how to lay out and organise arguments, which is the topic of this section.

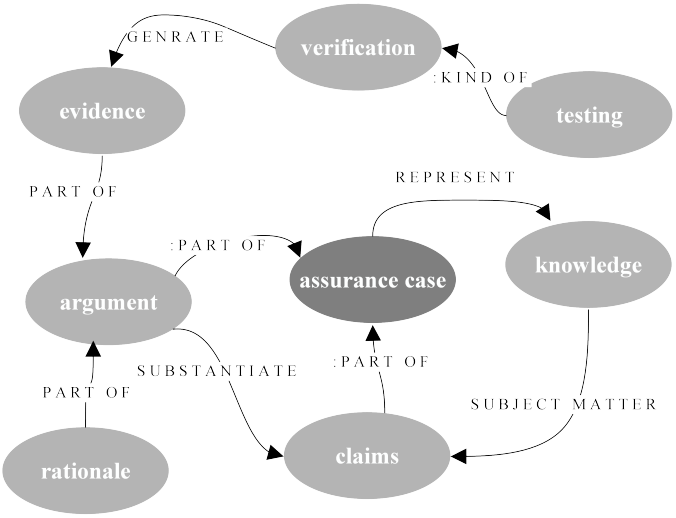


Figure 9. The assurance case represents the knowledge in an assurance effort.

One of the most recognised and influential argument schemas is the one described by Stephen Toulmin in his 1958 book ”The Uses of Arguments” [22]. Toulmin’s motivation was to create a richer format that better reflected how people argued in reality, rather than the more formal and traditional format consisting of premise and conclusion.

The argument layout consists of six elements [23]: Claim (or Conclusion) (C), Data (D) (or Datum, Toulmin uses both terms), Warrant (W), Qualifier (Q), Backing (B), Rebuttal (R) (Figure 10).

In the simplest form, (D) may be some evidence that proves that (C) is the case. The transition between (D) and (C) may not be trivial, so a warrant needs to act as an inference licence between (D) and (C); that is, (W) acts as a bridge between (D) and (C). (W) may also be challenged, so a backing (B) may be

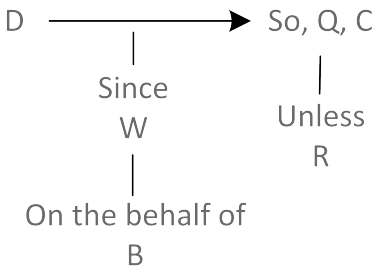


Figure 10. General layout of an argument [23, p. 97].

needed to support (W), that is, why (W) holds. (Q) indicates the strength of the step (i.e., strength of the ”bridge”) from (D) to (C). (R) indicates circumstances in which (W) may not hold.

Although the elements of an argument described by Toulmin are necessary aspects of an epistemic justification substantiating a proposition or assertion, the schema, in its simplest form, is insufficient for assurance of complex systems. The schema needs to be expanded.

Firstly, in the assurance of complex systems, there are many claims. System claims represent statements about the system properties and its use. These requirements address many system properties, including safety. Moreover, the claims must be refined at several levels of abstraction (LoAs). The LoAs link back to the LoAs connected to the systems approach and foundationalism.

Secondly, although one of Toulmin’s key motivations was to enable ”practical assessment of arguments” [23], he did not discuss aspects of argument assessment in detail. Clearly, when, e.g., a (top) claim is refined into two or more subclaims with accompanying justification, assessing the strength of each argument needs to be aggregated in some way to reflect the confidence in the top claim. Moreover, each element in the argumentation schema should be assessed, resulting in a network of assessments across different elements of an argument at different LoAs.

Several expanded argument schemas based on Toulmin have been developed, such as Goal Structuring notation (GSN) [24] and Trust-IT [25].

An assurance case organises these arguments systematically and in a structured manner, and represents the knowledge generated in the assurance (Figure 2). Different ways are possible based on the various argument schemas, such as [24] or [26]; both are compatible with [6]. A metamodel of an assurance case may also be found in [27].

IX. Stakeholder’s objectives and system requirements

Stakeholders hold objectives and pursue goals through utilising the system; that is, they use the system for a reason. A system’s mission is expressed as system requirements, which are derived from these objectives.

The stakeholders may be users, developers, and bystanders who have nothing to gain from the system but may be affected by it. Through its legislation and standards, the government represents stakeholders that cannot be consulted directly, such as the natural environment, future generations, the general public, children, etc. In such cases, conformance to standards means meeting stakeholders’ objectives and interests.

Stakeholders need confidence that their objectives are fulfilled or will be, or that a deviation from those objectives is acceptable. Implicitly, stakeholders also hold the objectives of being safe, secure, and treated fairly. These objectives may not be directly linked to the reason for developing and using the system in the first place (i.e., the mission). The system requirements must incorporate such implicit objectives. These kinds of system requirements can be termed mission-supporting requirements, or non-functional requirements [28], or even system constraints (Prof. Nancy Leveson terms this ”safety constraints”; however, when expanding the scope of such requirements to other system quality characteristics, they can be termed as ”system constraints”.) [7] (Figure 11).

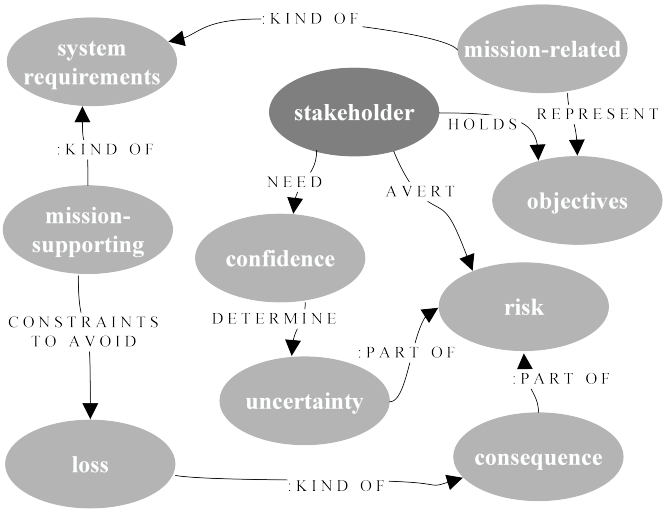


Figure 11. Stakeholders hold objectives that determine the system requirements.

In the context of assuring AI systems, mission-supporting system requirements should be based on a set of ethical principles, such as: [29].

Conflicts often arise between requirements directly related to the mission of the system and the mission-supporting requirements. Moreover, similar conflicts may also arise between the objectives and goals of different stakeholders, and even between different objectives of the same stakeholder (e.g., long-term vs. short-term goals). One understanding of ethics is: ”the identification, study, and resolution or mitigation of conflicts among competing values or goals” [30]. The assurance effort should document the trade-offs made between competing goals.

X. Conclusion and future work

This paper has presented a comprehensive ontology for the assurance of safety-critical systems, positing that assurance is fundamentally an epistemic activity. The core thesis establishes

knowledge as the central ”hub” of assurance; its systematic generation and explicit representation are the primary means of reducing epistemic uncertainty, which in turn builds justified stakeholder confidence. We have demonstrated how stakeholder concerns about potential loss are translated into system requirements and safety claims. These claims are substantiated by the assurance process, which generates knowledge structured and presented within an assurance case. The framework employs the CESM metamodel as a foundational systems approach to analyse the system behaviour and emergent properties, such as safety, that these claims address. Furthermore, we have introduced a multi-dimensional concept of objectivity as a critical metric for evaluating the strength of this knowledge. This metric ensures that the assurance effort is commensurate with the level of system risk.

This work shifts assurance from a traditional, compliance-focused approach to a more foundational and systematic methodology. This ontological model provides a scrutable and reasoned pathway for demonstrating a system's safety by clearly articulating the relationships between risk, knowledge, and confidence. This pathway is particularly significant for novel and complex systems, where established standards may be inadequate, and a deeper justification of safety is required to gain stakeholder acceptance. The framework offers practitioners a structured methodology to connect high-level stakeholder objectives directly to the evidence and arguments that form the basis of a safety case.

The scope is intentionally focused on assurance as an epistemic endeavour—the generation of knowledge to reduce epistemic uncertainty. Consequently, we did not discuss other vital risk management strategies in detail, such as altering system design to mitigate consequences or reduce aleatory uncertainty. Additionally, while the concept of objectivity provides guidance, experts must ultimately assess the strength of knowledge through a nuanced process that relies on judgement rather than a simplistic quantitative measure.

Building upon this foundation, further work could focus on operationalising the multi-dimensional objectivity metric into practical assessment tools. Practitioners could then apply the complete ontological framework to specific, challenging domains such as autonomous systems or artificial intelligence. A final valuable avenue for inquiry would be to investigate methods for aggregating argument strength across multiple levels of abstraction within a complex assurance case.

Ultimately, this paper provides a robust and coherent ontology for building justified confidence in the safety of complex systems. This approach offers a rigorous and defensible foundation for the responsible design, deployment, and operation of safety-critical complex systems by grounding assurance in the systematic generation of knowledge.

REFERENCES

[1] O. I. Haugen, ‘Integrating Assurance and Risk Management of Complex Systems’, in *2025 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Vienna, Austria: IEEE, 2025-10, pp. 6739–6746, ISBN: 979-8-3315-3358-8. DOI: 10.1109/SMC58881.2025.11342869. Accessed: 2026-04-13.

[2] DNV, *DNV-RP-0671 Assurance of AI-enabled systems*, Recommended Practice, 2023-09.

[3] International Organization for Standardization, *ISO/IEC/IEEE 31000 - Risk management*, International Standard, 2018-02. Accessed: 2026-04-13.

[4] C. R. Fox and G. Ülkümen, ‘Distinguishing Two Dimensions of Uncertainty’, in *Perspectives on Thinking, Judging, and Decision Making: A Tribute to Karl Halvor Teigen*, Universitetsforlaget, 2011, pp. 21–36, ISBN: 978-82-15-01878-2.

[5] D. Kahneman, *Thinking, Fast and Slow*, 1st ed. New York: Farrar, Straus and Giroux, 2011, ISBN: 978-0-374-27563-1 978-0-374-53355-7 978-0-606-27564-4.

[6] International Organization for Standardization, *ISO/IEC/IEEE 15026 Systems and software engineering–Systems and software assurance –Part 1: Concepts and vocabulary*, International Standard, 2019-03. DOI: 10.1109/IEEESTD.2019.8657410. Accessed: 2026-04-13.

[7] N. G. Leveson, *Engineering a Safer World: Systems Thinking Applied to Safety*. Cambridge, Massachusetts: MIT Press, 2012-01. DOI: 10.7551/mitpress/8179.001.0001. Accessed: 2022-10-21.

[8] M. Bunge, *Emergence and Convergence: Qualitative Novelty and the Unity of Knowledge* (Toronto Studies in Philosophy). Toronto ; Buffalo: University of Toronto Press, 2003, ISBN: 978-0-8020-8860-4.

[9] O. I. Haugen, ‘Safety assurance of complex systems Part 2: Assurance and analysis’, DNV AS, Høvik, Norway, Whitepaper, 2019. Accessed: 2026-04-13.

[10] O. I. Haugen, ‘A Systems Approach to Modelling Emergent Behaviour in Maritime Control Systems Using the Composition, Environment, Structure, and Mechanisms (CESM) Metamodel’, in *The Fifteenth International Conference on Performance, Safety and Robustness in Complex Systems and Applications*, vol. ISSN: 2308-3700, Nice, France: Think Mind, 2025-06, pp. 1–8, ISBN: 978-1-68558-280-7. Accessed: 2026-04-13.

[11] E. Hollnagel, *FRAM: The Functional Resonance Analysis Method, Modelling Complex Socio-Technical Systems*. Ashgate Publishing Limited, 2012.

[12] C. W. Bytheway, *FAST Creativity & Innovation: Rapidly Improving Processes, Product Development and Solving Complex Problems*. Fort Lauderdale, Fla: J. Ross Pub, 2007, ISBN: 978-1-932159-66-0.

[13] O. I. Haugen, ‘The Systems Approach’, in *Demonstrating Safety of Software-Dependent Systems; With Examples from Subsea Electric Technology*, T. Myhrvold and M. van der Meulen, Eds., DNV AS, 2022, pp. 145–163, ISBN: 978-82-515-0324-2. Accessed: 2026-04-13.

[14] O. I. Haugen, *An epistemic approach to confidence through objectivity in assurance of safety-critical complex systems*, 2024-12. Accessed: 2026-04-13.

[15] J. Nagel, *Knowledge: A Very Short Introduction* (Very Short Introductions 400), First edition. Oxford: Oxford University Press, 2014, ISBN: 978-0-19-966126-8.

[16] J. C. Watson, *Epistemic justification*, https://iep.utm.edu/epi-just/. Accessed: 2026-04-13.

[17] M. A. Falappa, G. Kern-Isberner and G. R. Simari, ‘Belief Revision and Argumentation Theory’, in *Argumentation in Artificial Intelligence*, I. Rahwan and G. R. Simari, Eds., 1st ed., Boston, MA: Springer Dordrecht Heidelberg, 2009-07, ISBN: 978-0-387-98196-3 978-0-387-98197-0. DOI: 10.1007/978-0-387-98197-0.

[18] F. Paglieri and C. Castelfranchi, ‘The Toulmin Test: Framing Argumentation within Belief Revision Theories’, in *Arguing on the Toulmin Model: New Essays in Argument Analysis and Evaluation*, D. Hitchcock and B. Verheij, Eds., Dordrecht: Springer Netherlands, 2006, pp. 359–377, ISBN: 978-1-4020-

4938-5. DOI: 10.1007/978-1-4020-4938-5_24. Accessed: 2023-11-06.

[19] F. Paglieri and C. Castelfranchi, *Argumentation and Data-oriented Belief Revision: On the Two-Sided Nature of Epistemic Change*, https://cmna.csc.liv.ac.uk//CMNA4/B.pdf, 2004-01. Accessed: 2026-04-13.

[20] H. E. Douglas, *Science, Policy, and the Value-Free Ideal*. University of Pittsburgh Press, 2009, ISBN: 978-0-8229-6026-3. DOI: 10.2307/j.ctt6wrc78. JSTOR: j.ctt6wrc78. Accessed: 2023-10-10.

[21] O. I. Haugen, 'Safety assurance of complex systems Part 3: Verification and evidence', DNV, Høvik, Norway, Whitepaper, 2019. Accessed: 2026-04-13.

[22] B. Verheij, 'The Toulmin Argument Model in Artificial Intelligence – Or: How semi-formal, defeasible argumentation schemes creep into logic', in *Argumentation in Artificial Intelligence*, 1st ed., Springer New York, NY, 2009-01, pp. 219–238.

[23] S. E. Toulmin, *The Uses of Argument*, 2nd ed. Cambridge University Press, 2002, ISBN: 978-0-521-53483-3.

[24] *Goal Structuring Notation Community Standard*, https://scsc.uk/gsn-standard, 2021-05. Accessed: 2026-04-13.

[25] J. Górski, Ł. Cyra, A. Jarzębowicz and J. Miler, *Argument Strategies and Patterns of the Trust-IT Framework*, 2008-01. Accessed: 2026-04-13. [Online]. Available: https://www.researchgate.net/publication/229034967%5C_Argument%5C_Strategies%5C_and%5C_Patterns%5C_of%5C_the%5C_Trust-IT%5C_Framework.

[26] *Argevide - System assurance management tools - Assurance cases*, https://www.argevide.com/home/, 2023-10. Accessed: 2024-01-15.

[27] *Structured Assurance Case Metamodel (SACM)*, https://www.omg.org/spec/SACM, 2023-10. Accessed: 2026-04-13.

[28] A. van Lamsweerde, *Requirements Engineering: From System Goals to UML Models to Software Specifications*. Chichester, England ; Hoboken, NJ: John Wiley, 2009, ISBN: 978-0-470-01270-3.

[29] High-Level Expert Group on AI, 'Ethics Guidelines for Trustworthy AI', European Commission, B-1049 Brussels, Tech. Rep., 2019-04. Accessed: 2022-06-10.

[30] L. McDaniel, *What Is Bioethics?*, https://bioethics.msu.edu/about/what-is-bioethics. Accessed: 2026-04-13.

Artificial Intelligence Contributions to Extending the Current Limitations of Virtual Reality for Integrating Operator Safety in Early-Stage Industrial Machinery Design

Rémy Houssin
ICube – CSIP University of Strasbourg
Strasbourg, France
Email: remy.houssin@unistra.fr

Amadou Coulibaly
ICube – CSIP INSA of Strasbourg
Strasbourg, France
Email: amadou.coulibaly@insa-strasbourg.fr

***Abstract*— Compliance with European standards now constitutes an essential for machine safety. Today, the integration of user safety is no longer regarded as a constraint, but as a function that the system must fulfill and as a responsibility shared by all stakeholders. However, the persistence of numerous accidents reveals the boundaries of a predominantly normative approach. This article presents an analysis of the contribution of Virtual Reality (VR) as a complementary tool for more effectively integrating human factors and operator safety from the design phase onward. First through a review of the literature, we demonstrate how VR made possible to anticipate working situations, reduce residual risks, and improve user-centered design. Subsequently, a prospective investigation is proposed to identify courant limitations of VR. Finally, we analyze the role of Artificial Intelligence (AI), which could potentially address and overcome these limitations.**

Keywords-User Safety; Virtual Reality; Design Process; Artificial Intelligence."

I. INTRODUCTION

User safety represents a big challenge in industrial systems. Despite the application of European directives and harmonized standards, statistics show a continued prevalence of occupational accidents. In 2023, the EU recorded approximately 2.83 million non-fatal workplace accidents, while the number of fatal accidents reached 3,298 [1]. These findings have led research efforts to focus on the early integration of human factors into design processes. In this context, VR has emerged as a key technology for overcoming the limitations of traditional approaches based only on standards [2]. The advent of AI has further strengthened the potential of VR, enabling more advanced prevention strategies and more accurate simulations of both technical and human behavior.

In Section 2 user safety integration in design phase is presented and risk assessment in Section 3. In Section 4 the contributions of VR are presented and its limitations are discussed in Section 5. AI potential contributions are proposed in Section 6 before concluding in Section 7.

II. INTEGRATION OF USER SAFETY FROM DESIGN PHASE

User safety is still not sufficiently integrated into the design process of mechanical systems [3]. It is often addressed late in the development cycle after the completion of CAD models, primarily to comply with the technical requirements of standards such as ISO 12100:2010 and EN 614.

European standards (Machinery Directive, ISO 12100, ISO 13849) provide a structured methodological framework for risk analysis. However, they are based on assumptions about use situations that are often idealized and imagined by designers. As a result, they struggle to represent real operator activity, adaptive strategies, human variability, and contextual constraints. These limitations partly explain the persistence of residual risks after compliance has been achieved.

Moreover, industrial machines are becoming increasingly complex, and it is difficult to account for all operating conditions related to the environment or actual use due to their high variability. This rapidly creates a gap between the way designers envision system use—based on various simulations, including those performed using VR—and the use actually emerges in practice as shown in Figure 1.

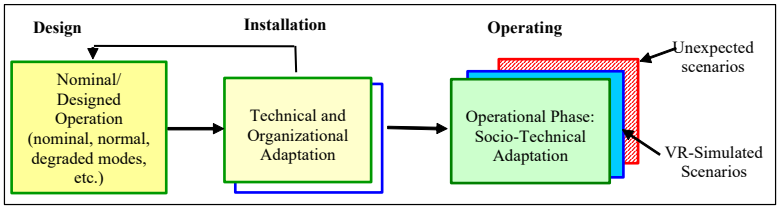


Figure 1. Evolution of the Gaps Between VR-Simulated Scenarios and the Real-Life System

Our involvement in this research problem aims to examine the impact that the integration of VR had on the design process of machines and industrial systems. Although standards are not static and evolve over time to better address designers' needs, and although VR can facilitate their application, we have observed that they are still predominantly applied during the final stages of the design process. This late integration tends to increase the complexity of the designed system

 ISBN: 978-1-68558-395-8

(through the addition of sensors, barriers, etc.) and requires cumbersome procedures, thereby increasing the operator's workload. Furthermore, existing standards insufficiently address technological hybridity within a single system, particularly as systems are likely to evolve throughout their life cycle as shown in Figure 2.

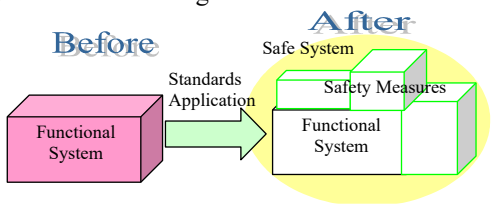


Figure 2. Current Integration of Standards Viewed as Increasing System Complexity

In general, a potentially hazardous phenomenon generated by a technical solution is delimited within a zone according to its nature. Depending on the phenomenon or phenomena present, this zone may be defined in terms of surface area or volume. This concept refers to any area inside and/or around a system in which a person is exposed to a risk of injury or adverse effects on health. Such a zone is generated, within a working situation [4], by a system or a component while performing a task or operating in idle mode.

The hazard zone can be identified at three levels:

- The most elementary level, corresponding to the technical solution that generates the hazardous phenomenon and zones.
- The system level, which represents the "assembly" dimension of technical solutions. At this level, hazardous zones defined at the first level may be modified or may even disappear. Figure 3 shows the effect of assembly options on the size and position of hazardous zones

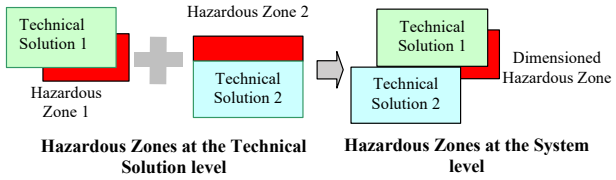


Figure 3. Dimensioned Hazardous Zones at the System Level

1. At the working situation level, the hazard zone does not exist prior to the installation of the system but results from the integration of the system at the site of use.

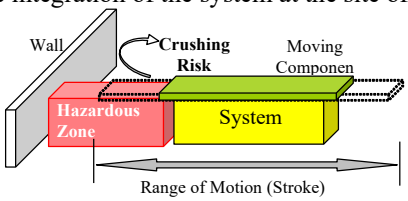


Figure 4. The Hazardous Zone in the Working Situation

Figure 4 shows installing a moving component in nearby a wall may create a hazardous zone due to the risk of crushing or trapping between the two systems, or between the moving component and the wall.

III. Risk Assessment

According to ISO 12100, risk Assessment is based on a combined analysis of, on one hand, the severity of potential harm and, on the other hand, the Frequency and duration of exposure to the hazard, as well as the technical and human Possibilities of avoiding harm. Figure 5 illustrates the parameters required for risk Assessment. Risk analysis and the selection of preventive measures have been detailed in [5]. The method enables designers and manufacturers to conduct risk analyses based on operators anticipated or foreseeable interventions, thereby identifying and implementing appropriate preventive measures for hazardous situations.

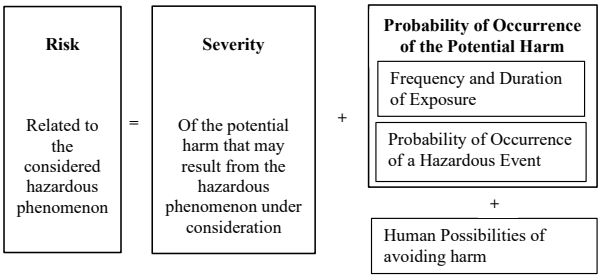


Figure 5. Risk Assessment Elements according to ISO 12100

The method suggests examining each operator task, operation by operation, in as much detail and concreteness as necessary. It involves listing the tasks and then performing a risk analysis by considering the various situations that could lead to harm. The designer is expected to anticipate resulting scenarios, including potential malfunctions, and to select appropriate preventive measures.

IV. Contributions of Virtual Reality

First of all, we should note that VR is not a normative method for risk assessment, but rather a tool that supports risk estimation, integrated into the design process of a machine or system.

VR has been widely used for the analysis and improvement of safety in industrial and intralogistics systems by integrating it into design, risk assessment, and training processes [3]. The authors clearly indicate that VR serves as a complementary or alternative technology to traditional approaches:

1. Enabling Immersive evaluation of hazards and countermeasures that are difficult to capture using standards alone.
2. Enabling the simulation of realistic working situations prior to machine manufacturing.
3. Offering the possibility of immersing operators, ergonomists, and designers in interactive virtual environments.
4. Facilitating the analysis of postures, gestures, movements, visibility, and accessibility.
5. Also allowing the identification of hazardous scenarios that are not anticipated by conventional normative risk analyses.

As confirmed by [6], for existing working situations or for the design of new situations that are highly similar to existing ones, VR enables the simulation of degraded conditions (fatigue, cognitive load, etc.), realistic contexts (noise, reduced visibility, etc.), rare or hazardous events (failures, human errors, etc.), and non-linear sequences of actions. Real operating conditions, which are often excluded from normative models, thus become observable, repeatable, and measurable.

Reference [7] demonstrates that virtual environments can be used to simulate operator activities while assessing postural constraints and a variety of operational strategies. This indicates that VR facilitates the examination of actual behaviors (gestures, postures, etc.) under constraints, which is critical for analyzing the gap between prescribed and real work and for detecting emerging risks.

A. VR role in design process,

When integrated into the design process, VR becomes an early validation tool. In [8], the authors analyzed the impact of VR on the design process and proposed guidelines for its integration. They emphasize that the potential of VR strongly depends on human-centered approaches and thoughtful methodological integration, rather than on systematic or purely technological adoption. VR enables the testing of different machine configurations, layouts, and procedures, including emergent behaviors and complex contextual situations. User feedback collected in virtual environments contributes to refining the design before costly physical prototyping phases. This approach reduces late design iterations and enhances the robustness of the design from a safety perspective.

VR also plays a significant role in modeling and simulating user behavior around the designed system. In certain cases, when the expected performance is not achieved, VR allows designers to question the initial design assumptions and to identify the need for modifications. While VR can influence and support the evolution of Computer-Aided Design (CAD) models, this influence is not automatic in all cases. Rather, VR primarily serves as an immersive evaluation and feedback tool that guides design changes. By immersing designers and users in full-scale 3D CAD models with natural interactions, VR makes ergonomic, safety, and accessibility issues visible—issues that are not always detectable through conventional screen-based reviews or 2D inspections.

VR enables the simulation of actual system use by incorporating human variability (body size, reach, etc.), co-activity with other operators, and realistic environments (lighting conditions, obstacles). Observations made in VR provide actionable insights that can be used to adjust CAD models prior to physical prototyping. So, issues identified in VR can be annotated and directly communicated to the design team, thereby creating a feedback loop toward the CAD model [9]. In some software environments, it is possible to generate collision markers, ergonomic constraints, or physical limitations. The authors in [10] noted that VR and digital human modeling are commonly employed within the context of Industry 4.0 for ergonomic assessment. However, their application remains limited when it comes to evaluating physical ergonomics across the various phases of product development. In their study, the authors propose a set of design guidelines that integrate VR and digital human modeling in order to anticipate physical ergonomics evaluations of assembly processes while the product is still under development.

B. VR role in risk assessment

VR also plays a significant role in risk assessment, notably through the following contributions:

1. Identification and characterization of hazardous situations: VR enables the simulation of normal, foreseeable, and degraded use scenarios by visualizing human–machine interactions. This makes it possible to reveal hazardous phenomena related to kinematics, accessibility, and actual operator gestures. As a result, VR enriches the identification of hazardous zone and hazardous situations, which constitutes a prerequisite for risk assessment. In [11], the authors found that VR-based risk assessment constitutes a robust and effective alternative to traditional document-based or CAD-based approaches. Although differences in hazard identification were observed between simple and more detailed virtual models, the overall risk evaluation outcomes remained largely comparable across model complexities. Based on their results, the authors recommend a progressive increase in model fidelity throughout the different phases of machine development, enabling risks to be identified in an economically and operationally efficient manner.
2. Contributions to risk assessment process:
   a) Severity of harm. VR helps to understand where and how the human body is exposed by visualizing impact, crushing, or shearing zones. It also allows comparisons of potential severity across different design concepts.
   b) Frequency and duration of exposure. VR makes it possible to simulate task repetitiveness and to observe the actual time operators spend in hazardous zones. Different operational scenarios (production, adjustment, maintenance) can be tested, leading to a more realistic assessment of human exposure to hazards.
   c) Possibility of avoidance of harm. One of the major contributions of VR concerns the evaluation of avoidance possibilities. By visualizing available reaction times, escape paths, and constrained postures or human reflexes, VR enables a concrete assessment of avoidance potential—an aspect that is often poorly evaluated using drawings or static representations.
3. Comparison of design concepts: VR allows multiple solution architectures to be compared in order to estimate which design generates fewer hazardous zones, reduces exposure, and facilitates avoidance.

Although VR does not directly produce a normative risk level, it provides observations and usage scenarios, either anticipated by designers or derived from real work situations involving similar systems or machines. These elements support technical arguments that are subsequently formalized

within the ISO 12100 risk assessment framework to guide design choices and risk reduction measures. Consequently, through the use of VR, risk assessment evolves from a purely documentary requirement into a decision-support tool for conceptual design.

V. CURRENT LIMITATIONS OF VR

Despite the strong contributions of VR to improving the performance of socio-technical systems, several recent scientific studies confirm that VR still faces substantial technical and economic barriers. These include limited hardware performance, high equipment and development costs, stringent technical requirements for integration and practical use, as well as constraints related to accessibility and maintenance. In the following, we discuss the current limitations of VR manifest at twelve levels detailed in the next 12 points :

1. Limitations related to VR itself and its use: [12] conducted a systematic review analyzing the use of VR (and Augmented Reality) for hazard detection and prevention. Their review highlights several methodological and technological limitations in current VR-based hazard recognition applications, including the lack of realistic scenarios, insufficient contextual learning, limited validation of results, restricted dynamic behavior of simulations, and persistent technological issues. These findings indicate that, although VR demonstrates strong potential in training, simulation, and design, such limitations continue to hinder its widespread adoption and generalization, particularly in industrial and safety-critical environments.
2. Limitations related to the maturity of design concepts: VR models used in early design phases are often incomplete or approximate. At upstream stages of the design process, kinematics are frequently hypothetical; velocities, masses, and inertias are not yet defined; and the materials and energy flows involved remain unknown or unquantified [13]. Under such conditions, VR may create an illusion of risk control, potentially masking unmodeled hazards such as vibrations, fatigue, or component failure. Consequently, while VR can make certain hazards visible, it cannot reliably quantify their severity, leading to possible underestimation of risk.
3. Limitations arise from the nature of risk assessment methods themselves: This is particularly evident in the qualitative treatment of harm severity, as real physical energies are often unknown during early design stages. The underlying physical dynamics and real-world energy levels are often unknown or insufficiently modeled at these stages [6]. The authors, demonstrate that although RV and virtual environments can be immersive and interactive, the fidelity of perception, movement, and physical relationships is constrained by current technologies. In other words, even when a hazardous situation is represented, the simulation does not necessarily reproduce the physical mechanisms, forces, energies and impacts that determine the actual severity of an accident. Additional challenges include the lack of realistic or contextualized scenarios and insufficient validation of results, which further limitations the reliability of VR-based risk assessment.
4. Limitations concern the integration of VR into the design process. During the early stages of product development, when such physical parameters are not yet fully defined or available. Strand [14] emphasizes that, VR environments often lack sufficient realism and functional depth to support detailed design tasks, particularly when accurate representation of physical interactions and energy levels is required. In [15], authors further argue that technical constraints directly influence how VR can be positioned within the design process, explaining why it remains complementary rather than substitutive. In [13], a systematic review of 49 papers emphasizes that although VR is used for certain design tasks, there is still no consensus or clear guidance on how to effectively apply VR during the conceptual design phase. This reflects the difficulty of handling immature or incomplete conceptual models in early design stages. They conclude that VR has not yet demonstrated a consistent and validated capability to manage such uncertainties. Similar conclusions are reported by [16], who highlight persistent challenges, including difficulties in assessing VR effectiveness during conceptual design, the lack of agreed-upon metrics for its use, and discrepancies between promising results and studies questioning its generalized effectiveness.
5. Limitations exist in capturing real work and actual user activity. Reference [17] describes an activity-centered ergonomic approach that distinguishes prescribed work from real work and emphasizes the analysis of real operational strategies. While this perspective strongly justifies the use of VR for activity analysis, it also highlights the difficulty of fully reproducing real gestures, adaptations, and decision-making processes in virtual environments.
6. Human and cognitive limitations affect VR-based evaluations. The perception of danger is subjective, and users may under- or overestimate risks depending on their expertise (expert versus novice). Extreme postures and compensatory strategies are often poorly anticipated. As a result, VR cannot guarantee homogeneous and objective risk evaluations. Reference [18] shows that VR alters functional body size perception and perceived distances due to sensorimotor distortions inherent to immersive systems (vergence, accommodation conflict). Reference [19] similarly demonstrates that spatial perception in VR is not strictly equivalent to reality, introducing depth and distance biases that can affect design decisions where spatial precision is critical.
7. The representation of the human body in VR is not neutral. Standardized anthropometries, idealized gestures, or the absence of real fatigue influence how users perceive hazards. In [20], the authors show that avatar morphology affects perceived affordances, while [21] demonstrates that manipulating virtual body size or shape (body ownership illusion) alters distance perception and motor judgments. Such biases have direct

implications for the evaluation of hazardous zones and safety distances.

8. Normative and regulatory limitations are substantial. VR cannot be considered proof of conformity, as standards such as ISO 12100 require formal analyses (FMEA, structured risk assessment, etc.). Consequently, VR results must be translated into normative frameworks. In [22], the authors analyze safety and privacy risks associated with XR technologies and argue for proactive regulation, highlighting the absence of harmonized standards addressing the intrinsic or extrinsic safety of VR systems in industrial contexts.
9. VR does not compute a regulatory risk level and cannot replace severity–frequency–avoidance matrices. It neither certifies compliance nor substitutes formal risk calculations. The European Parliament report on virtual worlds [23] underscores unresolved issues related to safety, responsibility, and legal frameworks. Reference [24] further shows that only a small proportion of VR safety studies rely on solid theoretical or normative foundations, and that standardized performance criteria and long-term evaluations are largely lacking.
10. Certification-related limitations remain critical. VR cannot be used to certify performance levels required by ISO 13849, as reliability calculations (MTTFd, DCavg, CCF) and documented validation activities remain indispensable. The standard does not recognize simulation alone as sufficient evidence for achieving required performance levels of safety-related control systems.
11. Economic and technical limitations constrain adoption, particularly for small and medium-sized enterprises. The combined costs of 3D modeling, usage scenario development, safety expertise, and VR implementation often result in limited return on investment. Technical challenges include limited interoperability between VR, CAD, and risk analysis tools, frequent updates, and the lack of automatic CAD model modification in most traditional systems. Although integrated platforms such as digital twins and immersive CAD environments (Siemens NX VR, Dassault 3DEXPERIENCE, Autodesk VRED) offer partial solutions. References [25] and [26] confirm that hardware limitations, high costs, interoperability issues, and user discomfort remain major obstacles.
12. Finally, immersive VR can have adverse effects on users. Reference [27] shows that even short immersive VR sessions can induce postural instability, disorientation, blurred vision, and nausea in vulnerable populations. More recent studies [28] and [29] report cybersickness, increased physiological stress markers, and decreased cognitive performance following prolonged VR exposure, indicating persistent effects that limit its use for precise or extended tasks.

In summary, a VR simulation constitutes a socio-technical system combining digital models, usage scenarios, and immersive human interaction to explore and analyze situations before they physically exist. While VR enables early anticipation of design and safety issues, it does not replace safety engineering practices or normative risk analysis, but rather complements them.

VI. ARTIFICIAL INTELLIGENCE CONTRIBUTIONS

VR has demonstrated significant potential for improving the integration of human factors and safety in the design of industrial machinery. However, as discussed in previous sections, VR presents several current limitations that reduce its effectiveness in early design phase and risk assessment. Artificial Intelligence (AI) offers targeted solutions to address these current limitations, enhancing VR's predictive, generative, and evaluative capabilities. Reference [30] highlights that, in high-risk training contexts, the integration of AI with VR significantly enhances the predictive, generative, and evaluative capabilities of VR-based simulations. So, AI could contribute to mitigate the ten key VR l current limitations cited above. In the following, we present the the nine identified potential contribution of IA.

1. Immature Models in the Conceptual Phase: To address models' limitations, AI could contribute to developpe predictive models based on machine learning and probabilistic reasoning. These models can estimate plausible ranges for velocities, masses, and inertias using historical databases of similar systems (case-based reasoning). Bayesian approaches could allow uncertainty to be explicitly modeled, generating risk envelopes rather than single-point estimates [31]. Additionally, AI could automate labor-intensive tasks such as scenario generation, collision detection, and postural analysis, reducing dependence on human expertise. These approaches enable a transition from deterministic simulations to informed, uncertainty-aware VR simulations, making VR scalable and more applicable in industrial contexts, including SMEs.
2. Difficulty in Quantifying Harm Severity: To address this boundary, AI Contribution manifest by coupling AI with biomechanical models allows the estimation of indirect harm indicators, such as probable impact zones, applied forces, and likely injury scenarios [32]. Learning from accident databases links geometric configurations to expected severity levels. So, AI could transform immersive observation into reasoned severity metrics, providing actionable data for risk evaluation.
3. Lack of Realistic and Contextualized Scenarios: Although VR scenarios may involve automated animations or scripted behaviors, they are most often manually designed at a conceptual level. The selection of situations, events, and interactions is typically driven by expert assumptions rather than generated autonomously by the system., failing to account for rare events, human errors, or co-activity, AI techniques such as reinforcement learning and automated planning can generate realistic, context-aware scenarios, including extreme or non-intuitive sequences that designers may overlook [33]. This could evolve VR from an illustrative tool to a generative and exploratory environment, expanding the scope of risk analysis.
4. Difficulty Integrating VR into the Design Process: Although guidelines exist for integrating VR into the

design process [8], industrial adoption remains heterogeneous. AI could function as a meta-decision layer, linking VR, CAD, and risk assessment workflows. Recommendation systems could suggest optimal VR interventions based on design maturity and development stage. So, AI could enable systematic integration of VR into the design process, enhancing coordination rather than supporting isolated applications.

5. Incomplete Representation of Real Work: While VR struggles to replicate real operator strategies and task variability, AI could learn from field data (videos, sensors, industrial log, etc.) to model real operator behaviors. Behavioral AI could simulate diverse profiles, accounting for expertise level, fatigue, and stress. So, VR simulations could become more accurately reflect real-world operations, bridging the gap between prescribed and actual work.
6. Human Subjectivity and Cognitive Biases: Hazard perception naturally varies across users, which can lead to inconsistent risk assessments in VR alone. By integrating AI, VR can be enhanced with quantitative indicators—such as exposure time, minimum distances, and joint angles—providing additional insights to support evaluation. AI can help identify potential perceptual biases and highlight critical aspects of operator behavior, strengthening the objectivity and reliability of VR-based assessments while complementing expert judgment.
7. Lack of Regulatory Recognition: VR alone cannot serve as proof of compliance with standards. However; AI could translate VR observations into structured, traceable normative arguments aligned with ISO 12100 and other regulations, supporting formal risk analyses. So, AI could act as a bridge between VR insights and regulatory frameworks, complementing but not replacing formal requirements.
8. Inability to Certify Safety Functions (ISO 13849): because VR cannot substitute for reliability calculations required for certification. the use of AI could prioritize critical safety functions for detailed analysis and detect risky system architectures early, guiding subsequent certification efforts to facilitate informed decision-making during certification planning without replacing formal compliance processes.
9. VR significant technical and economic burdens: Implementing VR in industrial design and safety assessment is resource-intensive, requiring substantial hardware, software, modeling, and expert personnel to create realistic scenarios, detect collisions, and evaluate operator postures. AI can mitigate these technical and economic burdens by automating time-consuming tasks, such as scenario generation, collision detection, and postural analysis. Machine learning and procedural generation methods can create a wide variety of realistic operational scenarios with minimal human input, while AI-driven algorithms can identify potential hazards automatically. This reduces reliance on highly specialized personnel and accelerates the evaluation workflow. As a result, VR becomes more scalable and industrially applicable, particularly for small and medium enterprises that have limited resources. By lowering labor costs and increasing the throughput of VR-based analyses, AI enhances the return on investment and facilitates broader adoption of VR for safety and design purposes.

These non-exhaustive identified potential contributions of AI need more explication about how they could be applied, to be verified an validated.

VII. CASE STUDY

This case study considers the early design phase of a semi-automated industrial packaging machine. The objective is to show how VR supports early safety assessment and how Artificial Intelligence (AI) can enhance current limitations.

Step 1: VR-Based Safety Assessment

A VR environment is created from preliminary CAD models, enabling immersive simulation of operator–machine interactions. Designers and safety engineers can explore operating and maintenance scenarios, identifying hazards such as collision risks, poor visibility, or non-ergonomic postures. This approach enables: early detection of safety and ergonomic issues, improved collaboration between stakeholders, and iterative design refinement at reduced cost. However, limitations persist: operator behaviors are often predefined, scenario coverage is limited, and risk assessment relies heavily on expert judgment.

Step 2: AI Contributions: in this case AI can be introduced to enhance VR-based safety assessment in three key areas:

1- Scenario Generation: Machine learning techniques can automatically generate diverse operating conditions, including rare or unexpected situations (e.g., abnormal machine states or emergency interventions). This improves the coverage of risk analysis beyond manually defined scenarios.

2- Behavioral Realism: AI models enable more realistic simulation of operator behavior by incorporating variability, reaction times, and non-ideal actions. This allows better representation of real-world human–machine interaction.

3- Risk Identification Support: AI can analyze interaction data within the VR environment to detect hazardous patterns, such as repeated proximity to dangerous components or unsafe sequences of actions. This supports systematic and data-driven risk identification.

Step 3: Linking VR and AI techniques: AI techniques, particularly in natural language processing, may assist in interpreting safety standards and linking them to design features. In this case, such tools could help identify applicable requirements and guide early design decisions, reducing the risk of non-compliance.

We could note that the combined use of VR and AI enables a more proactive approach to safety integration by shifting risk assessment to earlier design stages. Key benefits include improved hazard detection, increased scenario diversity, and enhanced decision support. However, challenges remain, including the need for high-quality

training data, integration into existing design workflows, and validation of AI outputs in safety-critical contexts. Therefore, the approach should be seen as complementary to traditional safety methods rather than a replacement. This case study demonstrates the potential of combining VR and AI to enhance early operator safety assessment in industrial machinery design. While conceptual, it provides a realistic illustration of how these technologies can improve current practices and supports future work toward implementation and validation.

VIII. CONCLUSION ET FUTURE WORK

VR represents a significant lever for enhancing the integration of human factors and operator safety in the design of industrial machinery. Complementing European standards, VR enables better anticipation of real work situations and reduces residual risks. While it does not replace regulatory frameworks, VR allows concrete verification of preventive measures defined through normative risk assessment and introduces an experimental, human-centered dimension that strengthens both functional compliance and user acceptability.

The systematic integration of VR into design processes offers a promising avenue for more proactive, user-centered safety. However, the limitations identified—technical, methodological, human, and regulatory—highlight the need for further research to mitigate these constraints. In particular, the integration of AI could enhance VR by generating and analyzing a wider range of usage scenarios, including hazardous situations, thereby supporting more comprehensive risk assessment.

Future developments combining VR, advanced digital models, and AI offer the potential for dynamic and predictive risk evaluation. By incorporating human variability and real operating conditions, these approaches can further improve accident prevention and reduce work-related musculoskeletal disorders at the earliest stages of design. In future work, such advancements will be positioned VR/AI as a critical tool that need evaluation and validation for the next generation of human-centric, safe industrial systems.

REFERENCES

[1] [Europa.eu 2025], "Rapport on virtual worlds – opportunities, risks and policy implications for the single market" https://www.europarl.europa.eu/doceo/document/A-9-2023-0397_EN.html, 2026.04.17.

[2] K. Lewczuk and P. Żuchowicz, "Virtual Reality Application for the Safety Improvement of Intralogistics Systems", Sustainability, vol. 16 issue. 14, 6024. doi.org/10.3390/su16146024, 2024

[3] L. Valentini, V. Weistroffer, F. Grandi and M. Peruzzini, "Digital toolkit for human-centered machine design: development and testing of an innovative system integrating virtual reality and HMI digital prototypes", *Int J Adv Manuf Technol*, 2025. doi.org/10.1007/s00170-025-16943-4.

[4] R. Hasan, A. Bernard, J. Ciccotelli and P. Martin, "Integrating safety into the design process: elements and concepts relative to the working situation", Safety Science. Vol. 41, Issues 2–3, 2003, pp 155-179, doi.org/10.1016/S0925-7535(02)00002-4.

[5] M. Compare, E. Zio, E. Moroni, G. Portinari and T. Zanini, "Development of a methodology for systematic analysis of risk reduction by protective measures in tyre production machinery", Safety Science, Vol. 110, Part A, 2018, Pages 13-28, doi.org/10.1016/j.ssci.2018.07.027.

[6] G. Personeni and A. Savescu, "Ecological validity of virtual reality simulations in workstation health and safety assessment", Front. Virtual Real., 16 February 2023 Sec. Virtual Reality in Industry, Vol. 4, 2023, doi.org/10.3389/frvir.2023.1058790.

[7] M. Zare, N. Bert, M. Norval, J. C. Sagot and A. Garrigou, "Diversité des stratégies opératoires en situation réelle et en environnement virtuel", Archives des Maladies Professionnelles et de l'Environnement, Vol. 86, Issue 3, 2025 doi.org/10.1016/j.admp.2025.102855.

[8] S. Stadler, and J. R. Chardonnet, "Embracing virtual reality: Empowering professionals in the design process", CRC Press, (2024), doi 10.1201/9781003306078-2

[9] A. Prinz, K. Y. Lee, A. Gupta, S. Park and D. Göhlich, "CAD in Virtual Reality: Integration of Solid Modeling in Standalone VR Application", Proceedings of the Design Society. 2025;5:831-840. doi:10.1017/pds.2025.10097.

[10] A. G. Silva, et al., "Design Guidelines for Combining Digital Human Modeling and Virtual Reality to Foresee Workplaces Ergonomics Issues During Product Development", Applied Sciences, 15(13), 2025, doi.org/10.3390/app15137083.

[11] P. Puschmann, T. Horlitz, V. Wittstock and A. Schütz, "Risk Analysis (Assessment) Using Virtual Reality Technology - Effects of Subjective Experience: An Experimental Study", Procedia CIRP, Vol. 50, 2016, Pages 490-495, doi.org/10.1016/j.procir.2016.04.115.

[12] T. Faiz, M. T. K. Tsun, A. Mahmud, and K. Y. A. Sim, "Scoping Review on Hazard Recognition and Prevention Using Augmented and Virtual Reality", Computers, 13(12), 307, 2024, doi.org/10.3390/computers13120307.

[13] T. Liao and J. She, "How does virtual reality (VR) facilitate design? A review of VR usage in early-stage engineering design", Proceedings of the Design Society. 2023, 3:2115-2124. doi:10.1017/pds.2023.212.

[14] I. Strand, "Virtual Reality in Design Processes: - a literature review of benefits, challenges, and potentials. Form Akademisk", vol. 13, issue 6, 2020, doi.org/10.7577/formakademisk.3874.

[15] F. V. De Freitas, M. V. M. Gomes and I. Winkler, "Benefits and Challenges of Virtual-Reality-Based Industrial Usability Testing and Design Reviews: A Patents Landscape and Literature Review", Applied Sciences, vol. 12 issue 3, 2022, doi.org/10.3390/app12031755

[16] A. Abughalia, C. Stechert, "A Decade of Virtual Reality in Product Development: A Literature Review of Effectiveness, Challenges, and Future Research", Procedia CIRP, Vol. 136, 2025, Pages 438-443, doi.org/10.1016/j.procir.2025.08.076.

[17] F. Barcellini, M. Cerf and M. Lacomblez, "Developmental foundations of Activity-Centered Ergonomics: knowledge encounters to construct both a critical analysis of work and developmental set-ups". Ergonomics. 68 (6): pp. 813–831, 2025, doi:10.1080/00140139.2024.2415965.

[18] X. M. Wang, A. Mazalek, C. M. Sabiston, T. N. Welsh, "Virtual Reality Alters Perceived Functional Body Size, Human-Computer Interaction", arXiv:2510.00824, 2025, doi.org/10.48550/arXiv.2510.00824.

[19] G. Kurpinar et al., "On the strengths and weaknesses of virtual reality in distance estimation in AEC domain: a meta-analysis of literature 2014–2024". Virtual Reality 30, 27, 2026, doi.org/10.1007/s10055-025-01276-0.

[20] T. Akkoc, E. Ugur, and I. Ayhan, "Trick the Body Trick the Mind: Avatar representation affects the perception of available action possibilities in Virtual Reality", Human-Computer Interaction, 2020, doi.org/10.48550/arXiv.2007.13048.

[21] H. Ryu and K. Seo, “The illusion of having a large virtual body biases action-specific perception in patients with mild cognitive impairment”. Sci Rep 11, 24058, 2021, doi.org/10.1038/s41598-021-03571-7.

[22] Hine, E. et al., “Safety and Privacy in Immersive Extended Reality: An Analysis and Policy Recommendations”. DISO vol.3, issue 33, 2024), doi.org/10.1007/s44206-024-00114-1

[23] Europarl 2023, Website, https://www.europarl.europa.eu/doceo/document/A-9-2023-0397_FR.html?utm_source=chatgpt.com. 2026.04.17.

[24] D. Scorgie, Z. Feng, D. Paes, F. Parisi and T. W. Yiu, “Lovreglio R., (2024) Virtual reality for safety training: A systematic literature review and meta-analysis”, Safety Science, Vol. 171, 2024, doi.org/10.1016/j.ssci.2023.106372.

[25] Y. Yang, “Technical Challenges Affecting the Popularization of Virtual Reality Technology”, Proceedings of International Conference on Mechanics, Electronics Engineering and Automation (ICMEEA 2024), Advances in Engineering Research 240, doi.org/10.2991/978-94-6463-518-8_26.

[26] A. D. Samala et al., “Virtual reality in education: global trends, challenges, and impacts game changer or passing trend?”. Discov Educ 4, 229 (2025). doi.org/10.1007/s44217-025-00650-z.

[27] M. Pau et al., “Cybersickness in People with Multiple Sclerosis Exposed to Immersive Virtual Reality. Bioengineering (Basel)”. 2024 Jan 24;11(2):115. doi: 10.3390/bioengineering11020115. PMID: 38391601; PMCID: PMC10886275.

[28] S. Vlahovic, L. Skorin-Kapov, M. Suznjevic, and N. Pavlin-Bernardic. “Not just cybersickness: short-term effects of popular VR game mechanics on physical discomfort and reaction time”, Virtual Reality 28, 108 (2024). doi.org/10.1007/s10055-024-01007-x.

[29] D. Zielasko, B. Rehling, B. Von Dawans, and G. Domes, “Do Not Immerse and Drive? Prolonged Effects of Cybersickness on Physiological Stress Markers And Cognitive Performance”, doi.org/10.48550/arXiv.2506.11536.

[30] P. Fernández-Arias, A. Del Bosque, G. Lampropoulos, and V. Vergara, “Applications of AI and VR in High-Risk Training Simulations: A Bibliometric Review”, *Applied Sciences*, *15*(10), 2025, 5424, doi.org/10.3390/app15105424.

[31] Y. Gal, P. Koumoutsakos, F. Lanusse, G. Loupe, C. Papadimitriou “Bayesian uncertainty quantification for machine-learned models in physics”, *Nat Rev Phys* **4**, 573–577, 2022, doi.org/10.1038/s42254-022-00498-4.

[32] Y. Jiang et al., “Analyzing Crash Severity: Human Injury Severity Prediction Method Based on Transformer Model”, *Vehicles*, *7*(1), 5, 2025, doi.org/10.3390/vehicles7010005.

[33] C. Lu, "Test Scenario Generation for Autonomous Driving Systems with Reinforcement Learning", 2023 IEEE/ACM 45th Proceedings (ICSE-Companion), Melbourne, Australia, 2023, pp. 317-319, doi: 10.1109/ICSE-Companion58688.2023.00086.

Using SORA Principles to Assess The Safety of Unmanned Traffic Management (UTM) Services

Sara Rachid
Thales Air Mobility Solutions (AMS)
Land and Air Systems (LAS),
Rungis, France
e-mail: sara.rachid@fr.thalesgroup.com

Fateh Kaakai
Thales Research and Technology (TRT),
Palaiseau, France
e-mail: fateh.kaakai@thalesgroup.com

***Abstract*—This paper presents a methodology to assess the safety of Unmanned Traffic Management (UTM) systems, not sufficiently addressed in the current state-of-the-art, using the concepts of the Specific Operations Risk Assessment methodology (i.e., SORA). This enables a smooth transition from traditional safety assessment frameworks to the new Unmanned Aircraft Systems (UAS) paradigms.**

Keywords- UTM; UAS; Safety; FHA; SORA.

I. INTRODUCTION

Operations executed with UAS are gaining more ground recently, with more complex airborne and ground systems and more diversified operational scenarios. This important growth of the UAS traffic implies, among other consequences, a growing safety risk that should be assessed and mitigated appropriately. For this purpose, several regulations were released around the world to define the rules and procedures required to ensure the safety of UAS operations. In addition, the SORA methodology was issued by JARUS (Joint Authorities for Rulemaking on Unmanned Systems) to provide UAS operators with a methodical framework to assess the safety of their operations. It can be observed, on the other hand, that the current state-of-the-art does not address sufficiently the safety of UTM services used to plan, deconflict, validate UAS flight requests prior to their execution, and also to launch the UA and ensure its safety during the flight. UTM services can be identified today as a real source of risk for UAS operations. In fact, such systems have a criticality proportionate to the safety risks of the UAS operations they support, as their malfunctioning can affect the safety of those operations. Consequently, the manufacturers of such systems should perform safety risk assessments in order to ensure the management of these risks. Yet, no adapted guidance is provided today to support this need. Concretely, UTM system manufacturers need to:

- determine the level of safety risk for their system,
- determine the level of development assurance (i.e., the level of industrial development rigor) for their system,
- and ensure a proportionality between the level of development assurance and the level of safety risk, in order to avoid oversized or undersized objectives of development assurance.

This can be challenging considering the operational specificities brought by the context of UAS operations, compared to traditional manned air traffic and other safety-related industries. Hence, traditional safety assessment methods (including non-specific methods) cannot be used directly and should be adapted to this new operational context.

To respond to this need, this paper proposes a methodology to assess the safety of UTM services and allocate commensurate safety risk levels according to the safety risks of the UAS operations they support. To achieve this goal, this paper integrates traditional safety assessment principles (e.g., Functional Hazard Analysis FHA) with the concepts of the SORA.

In the detail, the first contribution of this paper is the establishment of a severity matrix that defines the relevant operational aspects to evaluate (i.e., effect on people on the ground, manned aircraft and UAS crew), and the severity levels to consider for each of them. This step contributes to the specification of the operational environment of the system.

The second contribution of this paper consists of the development of transfer functions that associate the risks of UAS operations (represented by the SORA V2.5 risk metrics ARC, GRC and SAIL) to the commensurate severity levels. The resulting transfer functions will be used as part of the safety risk assessment of the system.

Finally, the last contribution of this paper consists of the validation of this methodology through its application on a real industrial use case.

As described, the proposed methodology is intended to enable a smooth transition from traditional safety risk assessment frameworks to the new UAS paradigms introduced by the SORA. In addition, this methodology helps UTM system constructors in fulfilling the regulatory requirement of demonstrating the safety of UTM systems, as in the regulation (EU) 2021/664 for instance.

In the following, the paper analyses first the state-of-the-art on the methods applicable to the context of UAS operations (in Section II), then describes the proposed methodology to assess the safety of UTM services (in Sections III and IV). An application of the methodology on an industrial case is presented after that (in Section V), in

addition to a discussion of its benefits and limitations (in Section VI).

II. STATE-OF-THE-ART ANALYSIS

Considering the various profiles of stakeholders related to UAS operations, various methods are available in the literature and regulations to answer their needs in terms of safety assessment. This section presents those methods and exposes their different intents and perspectives, in addition to discussing their potential applicability to the safety assessment of a UTM system.

A. SORA and Other Operation-Centred Risk Assessment Methods

The current regulations in The European Union (EU), The United States of America (USA) and other countries allocate to UAS operators the responsibility of demonstrating the safety of their operations, with the conduct of safety risk assessments that can be described as operation-centred for their scope and vision. The SORA is one of the few methodologies available currently in the literature that fill this need.

The SORA methodology is developed by JARUS, providing guidance for the UAS operator on how to evaluate and conduct a UAS operation in a safe manner. The European regulation (EU) 2019/947 presents the SORA as an acceptable means of compliance with Article 11 of the UAS Regulation (EU) 2019/947, which requires the conduct of safety risk assessments on UAS operations.

In accordance with its operation-centred perspective, the SORA aims at assessing the safety risks to which the UAS operation is exposed. For this, the SORA introduces the hazard of losing control over the UAS operation to represent its safety risk. The outcome of this hazard is related to the risk of a mid-air collision with a manned aircraft, the risk of a person struck on the ground and the risk of causing damage to critical infrastructure. While the risk of damage to critical infrastructure is not addressed by the SORA, the air and ground risks are represented respectively by The Air Risk Class (ARC) and The Ground Risk Class (GRC). The SORA starts with assessing the air and ground risks, in the case of a loss of control over the UAS operation using the ARC and GRC. When these are considered as unacceptable, the SORA defines a set of requirements to reduce the probability of losing control over the UAS operation. This is achieved by reducing the probability of malfunctions of a UAS operation that can cause this hazard. The design of the UA, human errors of the remote crew, operational procedures or external systems supporting the UAS operation are identified in the SORA as possible sources of this hazard and can be consequently subject to a set of requirements to robustify them.

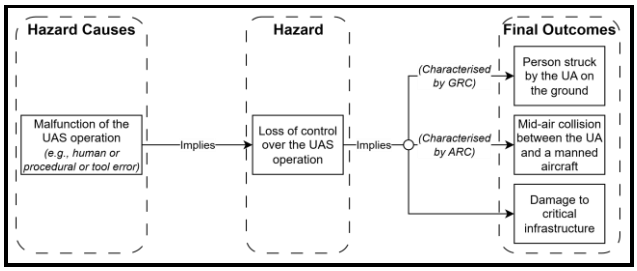


Figure 1. Risk Model for UAS Operations in The SORA

Figure 1. presents how these elements constitute the risk model for UAS operations of the SORA.

B. Airspace Risk Assessment Methods

The MEthoDology for the U-Space Safety Assessment (MEDUSA) is a method provided by the CORUS team (Concept of Operation for EuROpean UTM Systems) to address the need of assessing the safety of U-space airspaces. The MEDUSA relies on the SESAR Safety Reference Material (SRM) where safety assessment includes both a failure approach and a success one. In the context of U-space, this enables the assessment of U-Space's negative effect on the risk of an accident (failure approach), but also the positive contribution of U-Space to aviation safety (success approach).

The MEDUSA uses a holistic approach for the U-space safety assessment, incorporating both the operator and the airspace perspectives of U-space service provision, and the interoperability of these with manned Air Traffic Management (ATM). For this, the MEDUSA takes into account the outputs of the SORAs performed on the UAS operations expected in the U-space. These elements are integrated as a result in a single U-space safety assessment to obtain a unified airspace viewpoint.

The MEDUSA can also, on the other hand, recommend or require changes to be applied on the UAS operations expected in the assessed U-space, which results in changes in the SORAs.

More information on the MEDUSA is provided in [14] and [15].

C. Generic Risk Assessment Methods

To assess safety risks in more generic contexts, several methods are described in the literature and have been used traditionally for this purpose. Considering that the method presented in this article is applicable mainly to safety assessment methods that evaluate hazard severities, this section focuses on this type. As its title may indicate, this type of methods identifies failures at the level of the analysed scope and analyses their propagation and their potential effects and severities on the next higher level(s). Based on the targeted objectives of the analysis, these

methods can be employed on different levels of a system (e.g., piece-parts, functions, black-box, etc.).

In opposition to the MEDUSA and the SORA, traditional risk assessment methods are designed to be generic and applicable to different operational contexts.

In the context of safety assessments that evaluate hazard severities, the literature today contains mainly the Failure Mode and Effect Criticality Analysis (FMECA) proposed by the National Aeronautics and Space Administration (NASA) for space program hardware reliability [44], and the Functional Hazard Analysis (FHA) presented in the aeronautical safety process in ARP4761 [12].

D. Applicability of Risk Assessment Methods for UTM Systems

As afore explained, the intent of the SORA does not cover demonstrating the safety of UTM systems. It cannot therefore be used alone to fulfil this need, and a generic safety risk assessment should hence be conducted for this purpose. However, the SORA prescribes, when needed, a set of requirements on those systems and any other elements of the UAS operation to prevent a potential loss of control. Accordingly, to prevent the failures of UTM systems that may cause a loss of control, the SORA provides the Operational Safety Objective OSO#13: “External services supporting UAS operations are adequate for the operation”. This OSO should be met with a level of integrity (i.e., safety gain) and a level of assurance (i.e., method of proof), which both must be proportional to the risk of the UAS operation. Therefore, this OSO with its required levels of integrity and assurance are provided as inputs for the UAS operator to determine the need to perform a safety risk assessment on UTM systems.

On another level, the SORA presents an interesting method to assess the final outcomes of unsafe UAS operations in the air and ground, through the metrics ARC and GRC. These ones can be provided as inputs for the safety risk assessment of a UTM system. This use is possible assuming that the safety impact of such systems can vary based on the intrinsic risk of the UAS operations they manage.

In addition, it is important in the safety risk assessment of a UTM system to consider the operational assumptions and the mitigation means used in the SORAs of the UAS operations it manages. On one hand, this enables to retain the same ConOps studied in the operation SORAs and avoid thus inconsistencies between the SORAs and the safety risk assessments of UTM systems. On the other hand, the mitigation measures taken by the UAS operators (in their SORAs) can have a potential mitigation gain in the safety risk assessments of those systems.

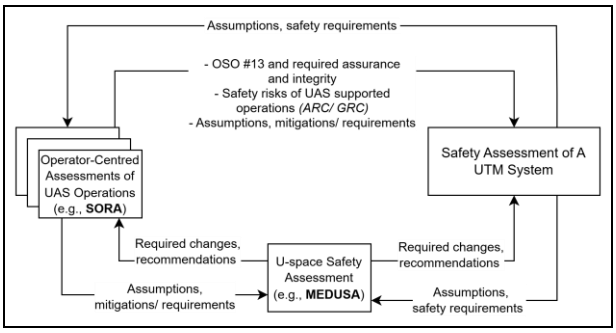


Figure 2. Link between The SORA and Safety Assessments of UTM Systems

As far as MEDUSA (and potentially other airspace safety assessment methods) is concerned, this assessment uses the outputs of SORAs to incorporate the operator perspective. In addition, if safety assessments of UTM systems in the assessed U-space are available, the MEDUSA is expected to take into account the results of those assessments. In fact, this enables to retain the operational use assumed for those systems, but also take advantage of the used mitigation measures. In the other way around, the MEDUSA may allocate requirements or recommendations to some UTM systems and/or to other operation elements. These would therefore result in potential changes to be considered in the SORAs and the safety assessments of the concerned systems.

Figure 2. represents the interfaces between the risk assessment methods, when applied in a context of UTM.

E. Summary of the State-of-The-Art

The UAS-related risk assessment methods available in the current state-of-the-art present little work on the safety assessment of UTM systems. Even less work is provided on methods that employ the concepts of SORA for assessing the safety risks of UTM systems. To address this need, this paper defines a safety methodology with the purpose of using the SORA concepts in the context of traditional safety risk assessment methods like FHA and FMECA.

III. METHODOLOGY RATIONALE

A. Impact of UTM Systems on UAS Operations

Various types of UTM systems can be distinguished, based on the operational phase of the UAS operation where they interfere. Some examples of such systems can be used in the pre-flight phase, i.e., before the execution of a UAS operation, notably for verifying the safety of that planned execution, authorising and preparing it. On the other hand, other systems can support UAS operations during their execution, i.e., in their in-flight phase, to ensure their safety.

Independently from the operational phase where such systems may interfere, the performed activities that rely on

them can have a potential impact on the execution of the concerned UAS operations. Consequently, the failures of those systems, while used to support UAS operations, can potentially have an impact on their execution as well.

In the concrete, if the failure of a UTM system can potentially, by definition, affect the execution of UAS operations and is not mitigated at an early stage, this one can result in an unsafe execution of the UAS operations that use the failing system service. An unsafe execution of a UAS operation takes place in the form of a direct exposition to its air and ground risks, with a compromised or annulled mitigating effect of the mitigation means that could reduce those risks.

In fact, when planning a UAS operation and assessing its safety, the UAS operator may eventually support it with one or many tools (e.g., operational procedures, systems) to reduce its estimated air and ground risks to an acceptable level. When a failure of a UTM system occurs, the efficiency of all the used mitigation means possessing dependencies with this one will be potentially impacted. In addition, if the occurring failure induces the execution of a UAS operation in conditions that are different from its initial plan, this can weaken the efficiency of several mitigation means or even devalue their use as their mitigating effect was assessed in accordance with the initially planned conditions. Therefore, a failure of a UTM system can compromise or annul the efficiency of those mitigation means, which implies that the UAS operations will be exposed to their initially unmitigated air and ground risk levels (see Figure 3.).

Based on that, the air and ground risk levels of a UAS operation are the characteristics to consider when assessing the safety impact of a failing UTM system, as it directly affects the severity of its outcome.

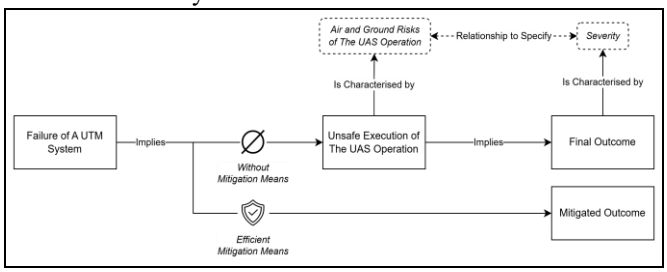


Figure 3. Propagation Scheme of Failures for UTM Systems

The ground risk of a UAS operation is the risk of a person on the ground struck by the UA, while the air risk of a UAS operation can be either the risk of a mid-air collision with a manned aircraft or with another UA. The risk of collision between UAs is excluded from the scope of the presented process. Consequently, the application of this process should be complemented with additional tools to cover this type of risk.

In what follows, the air risk refers to the risk of collision between a UA and a manned aircraft.

B. Use of SORA Concepts for UTM Systems Safety

It is hence clear that, in order to determine the appropriate severity level to associate to a similar scenario, it is first necessary to have an adequate method to define the air and ground risks and to determine their possible levels.

In this perspective, the ARC and GRC, introduced by the SORA, are estimated to provide a suitable representation of the air and ground risks to meet this need. As defined by the SORA, the ARC relates to the risk of a mid-air collision with a manned aircraft, and the GRC relates to the risk of a person on the ground struck by the UA (in the case of a loss of UAS control with a reasonable assumption of safety).

The ARC and GRC metrics and the matrices and procedures that are established to determine their values represent an efficient set of tools to express the variance of the level of risk of a UAS operation. In contrast to risk models based on continuous functions, the proposed ARC and GRC definitions furnish a finite number of risk classes covering the different possible risk degrees (i.e., low, medium and high risks). This discrete level distribution enables a simple association of ARC and GRC classes with severity levels. Furthermore, the ARC and GRC classes are defined using concrete parameters, which induces more facility to evaluate the safety impact represented by each ARC or GRC class and to associate it with the most convenient severity level. In addition, as the SORA is endorsed by the European Union Aviation Safety Agency (EASA) as an acceptable means of compliance with Article 11 of the UAS Regulation (EU) 2019/947, the ARC and GRC metrics are used for UAS risk assessments in different parts of the world, including the European Union (EU) and other countries that choose to follow European UAS regulations. Finally, these metrics are simple to use for they can be used independently from any software tools.

The cited elements make the ARC and GRC metrics a good choice to characterise the risks related to unsafe executions of UAS operations and to contribute to characterising the criticality of systems supporting them as a result.

C. Applicability of Methodology

As designed, the proposed methodology is applicable mainly to safety assessment methods that evaluate hazard severities. This includes methods like the FMECA and the FHA.

In addition, two types of risk related to UA operations will be addressed as part of this paper, in accordance with the SORA methodology. First, the “air risk” of a UAS operation will refer to the risk of a mid-air collision between

the UA and a manned aircraft. Secondly, the "ground risk" of a UAS operation will refer to the risk of causing physical harm to people on the ground, due to a collision with the UA. The risk of collision between UAs will be excluded from the scope of the presented process, and its potential safety consequences in the air and on the ground will not be considered in the assessment of the "air and ground risks" introduced earlier.

On another note, the « ground risk » in this article will not include the risk of a noise impact, privacy concerns or any other societal issues, nor the risk of damage to properties or critical infrastructure.

IV. PRESENTATION OF THE SAFETY ASSESSMENT METHODOLOGY FOR UTM SYSTEMS

A. *Definition of The Reference Severity Levels*

An unsafe execution of a UAS operation can impact the safety of operations on three levels. First, it can impact the capabilities of the UAS remote crew, by increasing their workload and/or debilitating their efficiency. It can also impact the safety of air traffic by interfering with manned aircraft using the airspace, and potentially causing reduction in safety margins with them which can go up to a mid-air collision. Finally, it can impact the safety of persons on the ground by causing events varying from a physical discomfort to one or multiple fatalities.

As a result, to determine the severity of an unsafe execution of a UAS operation, its final outcomes on the UAS remote crew, in the air and on the ground should be considered. Accordingly, the severity levels to be used in the context of UAS operations should be defined based on these three aspects. In alignment with this framework, the existing literature for the context of UAS operations rely on these safety aspects to provide definitions for severity levels.

TABLE I. TABLE OF REFERENCE SEVERITY LEVELS

Severity Category		Effect on People on The Ground	Effect on Manned Aircraft	Effect on UAS Crew
S5	No Safety Effect	Discomfort to persons	No safety effect	No safety effect
S4	Minor	Physical distress or minimal injuries to persons	Potential contingency manoeuvre to anticipate a reduction in safety separation, with no safety effect on the manned aircraft crew.	Slight increase in UAS crew workload, such as flight plan changes.
S3	Major	Non-serious injuries to persons	Significant reduction in safety separation between unmanned and manned aircraft	Significant increase in UAS crew workload or impaired remote crew efficiency
S2	Hazardous	Serious injuries to one or many persons, with no fatalities	Large reduction in safety separation between unmanned and manned aircraft	High workload such that the UAS crew cannot be relied upon to perform their tasks accurately or completely
S1	Catastrophic	Fatality or fatal injury to one or many persons	A collision with a manned aircraft	

To define the table of severity levels as part of the method presented in this article, two documents from the literature were chosen to be used as references: JARUS AMC RPAS.1309 Issue 2 and ATO Safety Management System Manual (SMS), December 2022. The consolidation of both severity tables results in TABLE I. , which will be used in the rest of the article.

B. *Association between SORA GRC and Severity Levels*

1) *Definition of The SORA GRC Metric*

The GRC metric is defined in the SORA *Step #2* "*Determination of the UAS intrinsic ground risk class (GRC)"* and the SORA *Step #3 Final GRC determination*. It relates to the risk of a person struck by the UA (in the case of a loss of UAS control with a reasonable assumption of safety). Its value is determined based on the maximum UA characteristic dimensions and/or its maximum speed, and the population density in the overflown ground area (see GRC table in the next section). The consideration of one or many external mitigation means could result in a reduced GRC value.

2) *Strategy of GRC and Severity Association*

To associate the GRC values to the appropriate severity levels, it is important to determine the safety impact presented by GRC values. In the literature, the fatality of striking a person on the ground by a UA (due to a loss of control over the operation) is represented in several existing workpapers as a function that varies based on the kinetic energy of the strike. Some proposed models may also involve the impacted regions of the human body (e.g., head, thorax, limb) as a second influencing parameter in this context.

In most of the existing studies, the event of striking a person on the ground, caused by an uncontrolled UA or other types of falling debris or fragments, causes a fatal untreatable injury (i.e., Abbreviated Injury Scale AIS = 6) and a probability of fatality of 90% when the kinetic energy absorbed by the human body surpasses 150 Joules (cf. Figure 4.). The curve providing this conclusion does not consider the variability of the fatality based on the impacted regions of the human body, as it is based on the *Average Body Position* data described in **Erreur ! Source du renvoi introuvable.**. In addition, the analysed events concern blunt force traumas caused by falling objects and excludes penetrating ones.

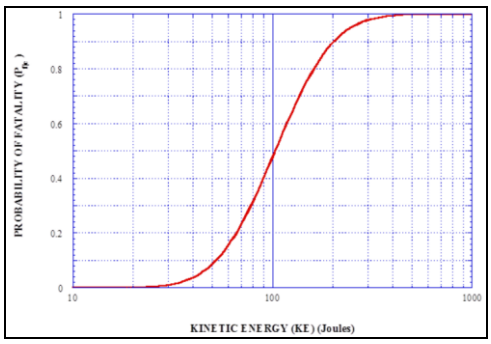


Figure 4. Kinetic Energy versus Probability of Fatality (Extract from [2])

The strike of a UA with a 25 m/s speed can generate a kinetic energy of 150 J if the UA weighs at least 480 grams. Several consumer UA models on the market with a 25 m/s speed exceed this weight and thus are expected to release more than 150 J, considered here as fatal. Based on that, it is considered that the first column of the table of GRC values (representing UAs with a 25 m/s speed, cf. TABLE II.) includes the risk of causing fatalities on the ground. Therefore, the GRC values contained in this column are representative of all the possible degrees of ground risk (i.e., low, medium, high). Consequently, the association between the GRC values and the appropriate severity levels can be done on this column first, before being extrapolated to the rest of the table. With this strategy, the associations will be based on a single variable: the population density on the ground.

In this sense, the first and second cells of the first column (i.e., GRC=1 and GRC=2) are considered to represent a negligible risk of having a person struck by the UA, as the probability of this event in controlled and remote ground areas is considered as negligible.

The third and fourth cells of the same column (i.e., GRC=3 and GRC=4), with a lightly and sparsely population density, are associated to a minor risk level. The fifth cell of the same column (i.e., GRC=5), related to suburban/ low density metropolitans, is related to a major risk level. The sixth cell of the same column (i.e., GRC=6), related to high density metropolitans, is related to a hazardous risk level. The last cell of this column (i.e., GRC=7), related to assemblies of people, is associated to the highest risk class, as it represents a high number of persons exposed to the UA and a limited ability of the persons to avoid it. The other values of GRC higher than 7 are all related to the highest risk level as well.

3) Transfer Function between GRC and Severity Levels:

Based on the ground risk division explained above, the GRC values are associated with the reference severity levels, which results in the transfer function provided by the following table:

TABLE II. TRANSFER FUNCTION BETWEEN GRC AND SEVERITY LEVELS

Max UAS Characteristics Dimension		1 m ≈ 3 ft	3 m ≈ 10 ft	8 m ≈ 25 ft	20 m ≈ 65 ft	40 m ≈ 130 ft
Maximum Speed		25 m/s	35 m/s	75 m/s	120 m/s	200 m/s
Maximum Population Density (people/ km²)	Controlled Ground Area	1 (S5)	1 (S5)	2 (S5)	3 (S4)	3 (S4)
	< 5 (Remote)	2 (S5)	3 (S4)	4 (S4)	5 (S3)	6 (S2)
	< 50 (Lightly populated)	3 (S4)	4 (S4)	5 (S3)	6 (S2)	7 (S1)
	< 500 (Sparsely populated/ Residential lightly populated)	4 (S4)	5 (S3)	6 (S2)	7 (S1)	8 (S1)
	< 5,000 (Suburban/ Low density metropolitan)	5 (S3)	6 (S2)	7 (S1)	8 (S1)	9 (S1)
	< 50,000 (High density metropolitan)	6 (S2)	7 (S1)	8 (S1)	9 (S1)	10 (S1)
	> 50,000 (Assemblies of people)	7 (S1)	8 (S1)	*Not part of SORA v2.5*		

C. Association between SORA ARC and Severity Levels

1) Definition of The SORA ARC Metric

As defined by the SORA V2.5, the ARC relates to the risk of a mid-air collision with a manned aircraft (in the case of a loss of UAS control with a reasonable assumption of safety). Its value is determined based the intrinsic characteristics of the used airspace (e.g., altitude, controlled versus uncontrolled airspace, etc.), in addition to the risk reduction potential of one or many strategic mitigation means.

2) Strategy of ARC and Severity Association

The ARC metric is defined in the SORA *Step #4 "Determination of the initial air risk class (ARC)"* and the SORA *Step #5 Application of strategic mitigations to determine the final ARC.* However, for the sake of simplification, the definitions of ARC classes used in this section are taken from the *SORA Step #6 "TMPR and robustness levels"* (see [5]). This SORA Step #6 provides generic and concise definitions of the ARC classes, involving both the airspace intrinsic characteristics and the impact of strategic mitigation means. In truth, the used definitions describe the ARC classes (i.e., ARC-a, ARC-b, ARC-c and ARC-d) based on two parameters: the probability of encountering manned A/C acquired from the intrinsic airspace characteristics, and the efficiency of the strategic mitigation means available.

To enable the transfer from the ARC classes to the reference severity levels as intended, a fifth ARC class needs to be added to this list. Considering the provided ARC definitions, the definition of the class ARC-c was estimated to be potentially dividable into two risk classes, to meet this need. The differentiation between the obtained risk classes, ARC-c and ARC-c', was set based on the same parameters used to define the other ARC classes.

3) Transfer Function between ARC and Severity Levels:

In the light of this, the resulting transfer function between the ARC classes and the severity levels is provided in TABLE III. and TABLE IV.

The association here between ARC-d and the catastrophic severity level (i.e., mid-air collision with a manned aircraft) assumes that no mitigations are available on the manned aircraft side to avoid a collision with a UA out of control. Therefore, the severity related to ARC-d can be reduced if we consider that manned aircraft can detect and avoid collisions with UAs.

TABLE III. TRANSFER FUNCTION BETWEEN ARC AND SEVERITY LEVELS

ARC-a (S5)	- Airspace where the **manned aircraft encounter rate is expected to be extremely low**.
ARC-b (S4)	- Airspace where the **likelihood of encountering another manned aircraft is low but not negligible** and/or where strategic mitigations address most of the risk and the **resulting residual collision risk is low**.
ARC-c (S3) / ARC-c' (S2)	- Airspace airspace with a **moderate likelihood of encounter with manned aircraft**, and/or where the **strategic mitigations available are medium robustness.**
ARC-d (S1)	**The manned aircraft encounter rate is high, and/or the available strategic mitigations are low.** Therefore, the resulting residual collision risk is high.

TABLE IV. MATRIX REPRESENTATION OF ARC CLASSES

Manned A/C Encounter Probability (Based on Intrinsic Airspace Characteristics)	Efficiency of Available Strategic Mitigation Means		
	Low	Medium	High
Extremely low	ARC-a (S5)	ARC-a (S5)	ARC-a (S5)
Low	ARC-b (S4)	ARC-b (S4)	ARC-b (S4)
Moderate	ARC-c'(S2)	ARC-c (S3)	ARC-b (S4)
High	ARC-d (S1)	ARC-c'(S2)	ARC-c (S3)

D. *Association between SORA SAIL and Severity Levels*

A Specific Assurance and Integrity Level (SAIL) is a parameter defined in the SORA as combination of the ARC and GRC levels of a UAS operation. This way, a SAIL is intended to reflect the level of confidence that the operation will remain under control, and to determine, when needed, the Operational Safety Objectives (OSOs) with their adequate levels of robustness to bring that level of confidence to an acceptable degree.

Having established earlier transfer functions to associate both ARC and GRC values to the appropriate severity levels, a transition from SAIL levels to severity levels can be defined as follows:

TABLE V. TRANSFER FUNCTION BETWEEN SAIL AND SEVERITY LEVELS

	Residual ARC				
Final GRC	a	b	c	c'	d
≤2	I (S5)	II (S4)	IV (S3)	IV' (S2)	VI (S1)
3	II (S4)	II (S4)	IV (S3)	IV' (S2)	VI (S1)
4	III (S4)	III (S4)	IV (S3)	IV' (S2)	VI (S1)
5	IV (S3)	IV (S3)	IV (S3)	IV' (S2)	VI (S1)
6	V (S2)	V (S2)	V (S2)	V (S2)	VI (S1)
7	VI (S1)	VI (S1)	VI (S1)	VI (S1)	VI (S1)
>7	Category C operation				

V. METHODOLOGY VALIDATION THROUGH INDUSTRIAL USE CASE

A. *Presentation of The Industrial Use Case*

The example used in this section is taken from a system that provides UA remote crews with the positions of manned aircraft in the vicinity of their UAs. The provided service is intended to support UA remote crews while operating their UAs, by contributing to their situational awareness and enabling them to detect and avoid potential mid-air collisions between their UAs and manned aircraft.

In the case of losing this service, the UA remote crew loses their ability to monitor manned traffic in proximity to their UA. Consequently, their ability to detect and avoid manned aircraft, which is considered as a main protection against the air risk, becomes impaired and/or completely inefficient. This is particularly impacting when the UAS operation is conducted in a BVLOS mode (i.e., Beyond Visual Line Of Sight). The described failure results hence in a direct exposition of the UA to the risk of a mid-air collision with manned aircraft.

To assess the safety impact of this failure on the operational environment, three independent propagation scenarios were analysed as shown in Figure 5.

The first scenario considers the case of UAS operations conducted in a VLOS mode. In this type of operations, the avoidance of mid-air collisions with manned aircraft is assumed to be fully achievable by the UA remote crew using their human vision only. Consequently, the safety of operations is not expected to be impacted by the loss of manned aircraft positions in the vicinity of their UAs.

The second scenario concerns BVLOS operations, where the use of external services for the provision of manned traffic positions has a significantly higher impact on the situational awareness of the remote crew. In this scenario, it is assumed that emergency recovery procedures (e.g., immediate landing procedures, Return To Home procedures, procedures for controlled crash, etc.) are available and can be applied by UAS remote crews to mitigate such cases. These procedures are generally defined by UAS operators and are intended to be used in “emergency” situations to ensure a safe termination of UAS operations. Based on that, this scenario will lead to protecting the UA against its operation air risk. However, the occurrence of this unexpected emergency situation may result in a physical discomfort of the UAS remote crew, which is related to a severity S4 in the table of reference severities.

The third scenario, also addressing BVLOS UAS operations, represents the case where no efficient emergency recovery procedures are available. This may be related to the absence of defined emergency recovery procedures to apply, the existence of defined emergency recovery procedures that are not efficient at ending safely UAS operations, or even the existence of defined emergency recovery procedures that have safety-impacting dependencies with the failing external service. In all the mentioned cases, the emergency recovery procedures are considered to have no mitigating effect on the air risk of the UAS operations. As a consequence, the UA will be exposed

 ISBN: 978-1-68558-395-8

directly to its operation ARC and the final outcome of this scenario will vary accordingly.

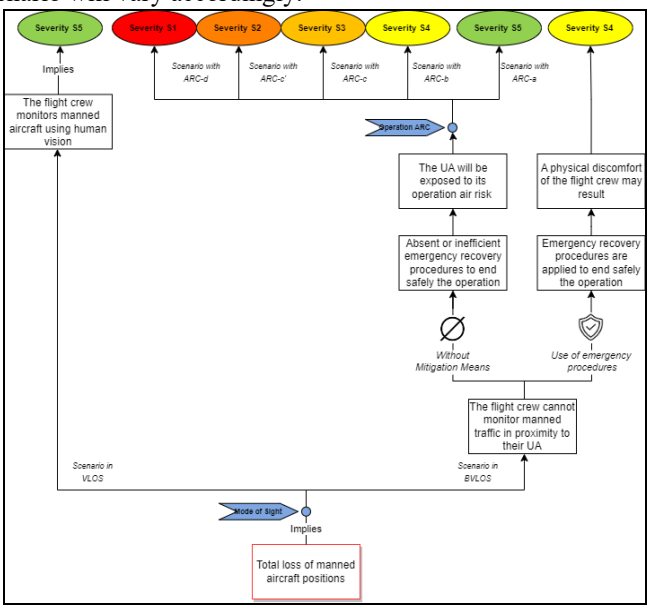


Figure 5. Graphical Representation of The Application Case

B. *Assumptions of The Analysis*

To illustrate efficiently the method presented in this paper, it is important to present a readable application case, that furnishes simple yet expressive scenarios, without including details that are irrelevant to this context. To achieve this vision, a set of assumptions were established to simplify the presented scenarios:

- It is assumed that a VLOS mode of sight is considered to be an acceptable tactical mitigation for collision risk for all ARC levels. This is supported by AMC 1, Article 11, Section 2.4.4.1 (a) in [5], applicable in the European context.
- It is assumed that the operational procedures and/or technical tools used by the UAS remote crew to mitigate the residual ground risk of their operation and to avoid collision with other UAs, do not rely on the failing service or have any other safety-impacting dependencies with this one that can affect their usability or efficiency.
- It is assumed that the operational procedures and/or technical tools used by the UAS remote crew to mitigate the residual ground risk of their operation and to avoid collision with other UAs, continue to be used to mitigate these risks during the execution of an emergency recovery procedure.
- It is assumed that the execution of an efficient or inefficient emergency recovery procedure cannot induce an increase of the ground risk.
- It is assumed that the execution of an inefficient emergency recovery procedure cannot induce an increase of the air risk.

VI. Conclusion and Future Work

As detailed above, this paper presents a methodology that uses the principles of the SORA to assess the safety of UTM systems. This is achieved through:

- the establishment of a severity matrix which defines a list of possible severity levels that can be achieved in the case of an unsafe execution of a UAS operation.
- The development of transfer functions that associates the risks of UAS operations (i.e., ground and air risks) to the commensurate severity levels.

This methodology is intended to help UTM system manufacturers in fulfilling the regulatory requirement of demonstrating the safety of UTM systems, as in the regulation (EU) 2021/664 for instance.

In fact, the provided transfer functions, associating ARC/GRC/SAIL metrics to severity levels, enable a straightforward conversion of these metrics into adequate final outcomes with determined severities. This represents a smooth integration of the SORA outputs into traditional safety risk assessment methods (e.g., Functional Hazard Analysis FHA). In addition, the employment of the SORA concepts here is intended to provide more harmonisation between the practices of system manufacturers and those of UAS operators, as the proposed methodology recalls the same concepts used on the UAS operator side. This will enable more interoperability between safety assessments produced on both sides. Also, the SORA brings the advantage of being a methodology already recognised and integrated as a UAS-specific safety risk assessment tool in several regions of the world.

It is important to note, when using the transfer functions, that the ARC/ GRC/ SAIL values used in the safety assessment of a UTM system shall have no safety-impacting dependencies with that UTM system. In fact, the SORA methodology enables the consideration of potential mitigations means for risk reduction, which influences the values of ARC/ GRC/ SAIL. On the other hand, the methodology proposed in this article uses the ARC/ GRC/ SAIL metrics to evaluate the safety risk related to a UTM system. Therefore, for a correct risk evaluation within the proposed method here, the ARC/ GRC / SAIL values shall not consider the UTM system as a risk reduction means.

On another level, the transfer functions between the SORA metrics (i.e., ARC, GRC, SAIL) and the end severities are established using expert-informed heuristics derived from domain expertise and the current state-of-the-art. This deliberate choice is considered as more compatible with the

operation-centric nature of the SORA. It also enables an easier understanding by safety practitioners and a simpler integration into their safety assessments. Iterative hardening is undoubtedly necessary in future enhancements of the methodology to move toward full industrial maturity. The growing industrial experience on UTM is also expected to help align the methodology with variables like the overall target level of safety for UAS operations, the assumptions made on the operational environment and the weight of operational parameters in the estimation of the end severities. These variables may be impacted due to the evolution of the operational framework of UTM, but also as a result of the growing return on experience on UTM operations.

As for the safety objectives that could be allocated to UTM systems based on each severity level, these are considered outside the scope of the proposed method. This choice is retained as the establishment of safety objectives for severity levels is difficult to generalise for all UTM systems. In fact, this relationship depends on several factors, including but not limited to: the Target Levels of Safety (also TLS), the contribution of UTM systems to the occurrence of undesired events and the complexity of UTM systems.

Moreover, this methodology does not address the risk of collision between UAs. Consequently, to perform an exhaustive safety assessment of a UTM system, the application of this methodology should be complemented with additional tools to cover this type of risk. One avenue for future work is to integrate the risk of collisions between UAs. This would allow UTM system manufacturers to cover the most significant safety risks associated with their systems through a single consistent methodology.

Ultimately, the deployment of the presented methodology in an industrial context is expected to be a gainful use on several aspects, when used within its applicability conditions.

REFERENCES

[1] Secretariat Range Commanders Council U.S. Army White Sands Missile Range, "RCC 321-00," Risk and Lethality Commonality Team Range Safety Group Range Commanders Council, NM, April 2000 Surname A and Surname B 2009 *Journal Name* **23** 544.

[2] Department of Defense Explosives Safety Board, "Procedures for the Collection, Analysis, and Interpretation of Explosion-Produced Debris", DDESB TP 21, Dec. 2007.

[3] Z. Svatý, L. Nouzovský, T. Micunek and M. Frydrýn, "Evaluation of The Drone-human Collision Consequences," Heliyon, vol. 8, issue 11, Nov. 2022, doi:10.1016/j.heliyon.2022.e11677.

[4] Í. de Oliveira, J. Fregnani, G. Balvedi, M. Ulrey and J. Musiak, "Safety Analysis Methods for Complex Systems in Aviation," The fifteenth Air Transportation Symposium (XV SITRAER 2016), Nov. 2016, arXiv:2208.02018.

[5] EASA. "Easy Access Rules for Unmanned Aircraft Systems," Jul. 2024. [Online]. Available from https://www.easa.europa.eu/.

[6] P. Stastny and A. Stoica, "Safety Management for Unmanned Aviation," INCAS BULLETIN, vol. 13, issue 2021, pp. 213 – 228, doi:10.13111/2066-8201.2021.13.4.18.

[7] JARUS. "JARUS Guidelines on Specific Operations Risk Assessment (SORA)," ed. 2.5, May. 2024. [Online]. Available from: http://jarus-rpas.org/.

[8] Anamta Khan, "Risk Assessment, Prediction, and Avoidance of Collision in Autonomous Drones," The 17th European Dependable Computing Conference (EDCC 2021), Sep. 2021, arXiv:2108.12770.

[9] EASA. "Commission Implementing Regulation (EU) 2019/947 of 24 May 2019 on the rules and procedures for the operation of unmanned aircraft," May. 2019. [Online]. Available from: https://eur-lex.europa.eu/.

[10] EASA. "Commission Implementing Regulation (EU) 2021/664 of 22 April 2021 on a regulatory framework for the U-space," Apr. 2021. [Online]. Available from: https://eur-lex.europa.eu/.

[11] EASA. "Acceptable Means of Compliance and Guidance Material to Regulation (EU) 2021/664 on a regulatory framework for the U-space," May 2024. [Online]. Available from https://www.easa.europa.eu/.

[12] SAE International. "ARP4761, Guidelines and Methods for Conducting The Safety Assessment Process on Civil Airborne Systems and Equipment," issue 1996-12, 1996.

[13] J. Qin, Y. Xi and W. Pedrycz, "Failure mode and effects analysis (FMEA) for risk assessment based on interval type-2 fuzzy evidential reasoning method," Applied Soft Computing, Volume 89, April 2020, 106134.

[14] SESAR Joint Undertaking, "Intermediate ConOps Annex D MEthoDology for the U-Space Safety Assessment (MEDUSA)," CORUS consortium, Ed. 01.00.00, 2019.

[15] SESAR Joint Undertaking, "U-space Concept of Operations," CORUS consortium, Ed. 03.00.02, 2019.

[16] K. ZAŁĘSKI, "Unmanned Aircraft as A Growing Hazard for Aviation Safety," Modern Management Review, vol. 13, 25 (2/2018), p. 99-111, Apr. 2018.

[17] L. Sedov, V. Polishchuk, T. Maury, M. Ulloa and D. Lykova, "Qualitative and Quantitative Risk Assessment of Urban Airspace Operations," Proc. The 11th SESAR Innovation Days, Dec. 2021.

[18] R.V. Melnyk, D.P. Schrage, V.Volovoi, and H. Jimenez, "Develop Sense and Avoid Requirements for Unmanned Aircraft Systems Using a Target Level of Safety Approach," Risk Analysis, Oct. 2014, doi:10.1111/risa.12200.

[19] J. Stevenson, S. O'Young and L. Rolland, "Estimated levels of safety for small unmanned aerial vehicles and risk mitigation strategies," Journal of Unmanned Vehicle Systems, vol. 3, issue 4, pp. 205-221, Sep. 2015.

[20] SESAR Joint Undertaking and Eurocontrol, "Guidance to Apply SESAR Safety Reference Material," 2018.

[21] A. La Cour-Harbo, H. Schiøler, " Probability of Low-Altitude Midair Collision Between General Aviation and

Unmanned Aircraft," Risk Analysis, vol. 2019, issue 39, pp. 2499–2513.

[22] The MITRE Corporation, "Modeling Risk-Based Approach for Small Unmanned Aircraft Systems," 2018. [Online]. Available from: https://www.mitre.org/.

[23] A. La Cour-Harbo, "Quantifying risk of ground impact fatalities of power line inspection BVLOS flight with small unmanned aircraft," International Conference on Unmanned Aircraft Systems (ICUAS), IEEE, Jun. 2017, pp. 1352-1360, ISBN: 978-1-5090-4495-5.

[24] S. Oh, Y. Yoon and S. Kim, "Risk Analysis of Unmanned Aerial System Operations in Urban Airspace Considering Spatiotemporal Population Dynamics," 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), IEEE, Nov. 2022, pp. 428-433, ISBN: 978-1-6654-6880-0.

[25] A. Allouch, A. Koubâa, M. Khalgui and T. Abbes, "Qualitative and Quantitative Risk Analysis and Safety Assessment of Unmanned Aerial Vehicles Missions Over the Internet," in IEEE Access, May 2019, vol. 7, pp. 53392-53410, doi: 10.1109/ACCESS.2019.2911980.

[26] H. E. Roland and B. Moriarty, "System Safety Engineering and Management", 2nd Ed., John Wiley & Sons, Inc., 1990.

[27] R. Clothier and R. Walker, "The Safety Risk Management of Unmanned Aircraft Systems", Handbook of Unmanned Aerial Vehicles, pp 2229–2275, Jan. 2014.

[28] L. Speijker, D. Ozuncer, J.A. Stoop and R. Curran, "Development of a Safety Assessment Methodology for the Risk of Collision of an Unmanned Aircraft System with the Ground", SAE Technical Papers, Oct. 2011, doi:10.4271/2011-01-2684.

[29] JARUS, "AMC RPAS.1309 Issue 2, Safety Assessment of Remotely Piloted Aircraft Systems", Nov. 2015. [Online]. Available from: Available from: http://jarus-rpas.org/.

[30] EUROCAE, "ED-279, Generic Functional Hazard Assessment (FHA) for UAS and RPAS", Oct. 2020.

[31] FAA, "49 USC 44809, Exception for limited recreational operations of unmanned aircraft", Oct. 2018.

[32] FAA, "14 CFR Part 107, Small Unmanned Aircraft Systems", Jun. 2016.

[33] FAA, "14 CFR Part 91, General Operating and Flight Rules".

[34] FAA, "FAA Order 8040.6, Unmanned Aircraft Systems Safety Risk Management Policy", Apr. 2019.

[35] EASA, "Certification Specifications and Acceptable Means of Compliance for Large Aeroplanes (CS-25), Amendment 27", Nov. 2021.

[36] FAA ATO, "ATO Safety Management System Manual (SMS)", December 2022.

[37] IEC, "IEC 61508, Functional Safety of Electrical/ Electronic/ Programmable Electronic Safety-related Systems".

[38] N. Leveson, "Engineering a safer world: Systems thinking applied to safety", in MIT Press, 2012, ISBN: 9780262298247.

[39] S. Du, G. Zhong, F. Wang, B. Pang, H. Zhang and Q. Jiao, "Safety risk modelling and assessment of civil unmanned aircraft system operations: A comprehensive review", in Drones, vol. 8, issue 8, May 2024. Retrieved from https://www.mdpi.com/2504-446X/8/8/354.

[40] J. Stádník, Š. Hulínská and J. Kraus, "Comparison of methods for the safety evaluation of UAS operation", Transportation Research Procedia, vol. 65, pp. 621-628, Nov. 2022. Retrieved from https://www.sciencedirect.com/science/article/pii/S2352146522006779/pdf.

[41] L. Meyer, C. Carlsson, Å. Svensson and M; Peukert, " Stressing safety assessment methods by higher levels of automation", Proc. The 33rd International Congress of the Aeronautical Sciences (ICAS), Sep. 2022. Retrieved from https://www.icas.org/ICAS_ARCHIVE/ICAS2022/data/papers/ICAS2022_0903_paper.pdf.

[42] F. Bonfante, "Safety Management System for Light RPAS", in CORE, 2020. Retrieved from https://core.ac.uk/download/pdf/234931050.pdf.

[43] E. Stefana, G. Di Gravio, R. Patriarca and F. Costantino, "Adopting the specific operations risk assessment methodology for drone inspections at industrial sites", "7th International Conference on System Reliability and Safety (ICSRS 2023), IEEE. Retrieved from https://ieeexplore.ieee.org/abstract/document/10381275/.

[44] R. Borgovini, S. Pemberton and M. Rossi, "Failure Modes, Effects and Criticality Analysis (FMECA)", Reliability Analysis Center, 1993.

[45] A. Shoufan, R. Alkadi, "Integrating Counter-UAS Systems into the UTM System for Reliable Decision Making", Nov. 2021, arXiv:2111.07291.

[46] K. Spalas, "Towards the Unmanned Aerial Vehicle Traffic Management Systems (UTMs): Security Risks and Challenges", Aug. 2024, arXiv:2408.11125.

[47] M. Rubagotti, I. Tusseyeva, S. Baltabayeva, D. Summers and A. Sandygulova, "Perceived Safety in Physical Human Robot Interaction - A Survey", May 2021, arXiv:2105.14499v1.

[48] J. Xiang, J. Xie and J. Chen, "Learning-accelerated A* Search for Risk-aware Path Planning", Sep. 2024, arXiv:2409.11634v1.

[49] S. Pohland and C. Tomlin, (2024). "PaRCE: Probabilistic and Reconstruction-Based Competency Estimation for Safe Navigation Under Perception Uncertainty", Sep. 2024, arXiv:2409.06111v1.

[50] B. Clement, M. E. Dubromel, P. Santos, K. Sammut, M. Oppert and F. Dayoub, "Hybrid Navigation Acceptability and Safety", Apr. 2024, arXiv:2404.11882v1.

[51] T. Savas, "A Risk-Based Analysis of Lightweight Drones: Evaluating the Harmless Threshold Through Human-Centered Safety Criteria", in Drones, vol. 9, issue 8, Jul. 2025, https://doi.org/10.3390/drones9080517.

[52] S. A. H. Mohsan, M. A. Khan, F. Noor, I. Ullah and M. H. Alsharif, "Towards the Unmanned Aerial Vehicles (UAVs): A Comprehensive Review" in Drones, vol. 6, issue 6, Jun. 2022, https://doi.org/10.3390/drones6060147.