A Novel High-Level Reasoning Architecture for Aircraft Prescriptive Maintenance

Haroun El Mir[1], Fakhre Ali[1], Ian Jennions[1], Steve King[1], Martin Skote[1], Lusitha Ramachandra[1]

[1]*Faculty of Engineering and Applied Sciences, Cranfield University, Bedfordshire, MK43 0AL, United Kingdom*

h.el-mir@cranfield.ac.uk
f.ali@cranfield.ac.uk
i.jennions@cranfield.ac.uk
s.p.king@cranfield.ac.uk
m.skote@cranfield.ac.uk
lusitha.ramachandra@cranfield.ac.uk

ABSTRACT

State-of-the-art Integrated Vehicle Health Management (IVHM) systems and digital twins (DTs) integrate physics-based and data-driven methodologies for predictive maintenance. Such systems commonly incorporate multiple DT instances to estimate substantive outputs. Nonetheless, they exhibit key limitations: lack of multi-DT orchestration mechanisms, limited uncertainty quantification, and insufficient prescriptive decision-support capability. To this end, this paper introduces an innovative High-Level Reasoner (HLR) decision support architecture for aircraft systems. The proposed HLR architecture comprises a multi-layer data-transfer structure, with the principal HLR layer consisting of adaptable specialist modules that facilitate a robust decision support implementation for query-driven prescriptive maintenance. The developed architecture is illustrated on an aircraft landing gear system (ATA 32), orchestrating multiple federated subsystems; represented by the Brake Temperature DT and Tyre Pressure DT. The contribution is a modular architecture that provides a reusable framework for extension across aircraft systems and wider IVHM applications. It therefore serves as an enabling technology that advances beyond existing diagnostics and prognostics solutions for asset utilisation.

1. INTRODUCTION

1.1. Background and Context

Commercial aviation maintenance operates under sustained cost and availability pressure. Oliver Wyman's 2024-2034 fleet and Maintenance, Repair, and Overhaul (MRO) forecast projects the global commercial-aircraft MRO market growth at USD 124 billion by 2034 (Prentice et al., 2023). In parallel, recent systematic review evidence indicates that artificial intelligence research in aircraft maintenance is concentrated in airworthiness management, aircraft health monitoring, and MRO operations rather than in isolated algorithmic studies (Pavlyuk & Alomar, 2026). Against this backdrop, predictive-maintenance capability has matured in industrial practice, but its implementation remains functionally distributed. Platforms such as Lufthansa Technik's AVIATAR , Air France Industries-KLM's PROGNOS, and Honeywell Forge Connected Maintenance are oriented toward airline and MRO operations, combining aircraft data ingestion with health monitoring, event prediction, troubleshooting, and operational maintenance support (Anandavel et al., 2021; El Mir et al., 2023; Shukla et al., 2020). By contrast, services such as Boeing Aircraft Health Management ("Health Management System for New 777-300ERs," 2006), GE Aerospace's Event Measurement System (EMS) and Maintenance Insight (Butter, 2019), Pratt & Whitney's FAST/EngineWise (Corrêa Macedo et al., 2025), MTU Maintenance's ETM (Brandt, 2017), CFM International's LEAP health-monitoring tools (Bastard et al., 2016; Forest, 2021), and Safran Landing Systems' predictive-maintenance services (Skaltsis et al., 2025) are more closely aligned with specific product families or subsystem domains, and accordingly offer greater depth in diagnostics and prognostics within those boundaries. Airbus Skywise (Bernard & Hoffmann, 2023) and Rolls-Royce's Blue Data Thread (Elliott, 2021) extend the industrial landscape further by providing digital environments for data aggregation, application deployment, and stakeholder connectivity. However, these environments do not in themselves constitute an aircraft-level prescriptive reasoning architecture. As noted by El Mir et al., maintenance-relevant data remains fragmented across OEMs, operators, and MRO organisations, with limited interoperability between stakeholder-owned data environments (El Mir et al., 2023).

The principal limitation, therefore, is not the absence of predictive-maintenance methods, but the absence of a system-level reasoning and orchestration mechanism capable of integrating fragmented functional capabilities. Existing developments are typically distributed across subsystem-level diagnosis and prognosis, platform-level data infrastructures, semantic integration mechanisms, and planning or optimisation modules. However, the literature still lacks a unifying framework capable of receiving a maintenance query (Kabashkin, 2025; Norcaro et al., 2025), identifying the relevant heterogeneous models and algorithms, orchestrating their execution across interdependent aircraft subsystems, and producing a traceable prescriptive recommendation.

1.2. Problem Statement and Research Objective

Prescriptive maintenance requires a higher degree of integration than diagnosis or prognosis considered independently. Diagnosis seeks to identify the fault hypotheses most consistent with available observations, whereas prognosis estimates degradation progression and the time to reach a defined threshold or failure condition. Prescriptive maintenance, by contrast, must transform these health estimates into recommended maintenance actions under resource, operational, and scheduling constraints. Prior research on aircraft maintenance-planning has highlighted the importance of this step by examining the influence of condition-monitoring information on service waiting times, maintenance timing, and fleet utilisation (Meissner et al., 2021).

To address this gap, this work defines and implements a modular high-level reasoning architecture positioned above subsystem digital twins and associated engineering algorithms. The architecture accepts structured maintenance questions, maps them to executable query types, identifies the relevant expert modules, orchestrates their execution, and integrates the resulting evidence into a traceable prescriptive recommendation. Its purpose is not to replace subsystem models or maintenance-planning tools, rather to coordinate them through a query-driven reasoning process. In this sense, the HLR is fundamentally an architecture, with its modular design providing a reusable framework for action-oriented maintenance decision support.

2. LITERATURE REVIEW

The development of Integrated Vehicle Health Management (IVHM) architectures in aerospace to date has been led by diagnostic problem-solving rather than prescriptive maintenance orchestration. To this end, the literature is reviewed here in a structured sequence to position the present work. Section 2.1 examines diagnostic, prognostic, and platform-level IVHM architectures, which establish the development of health reasoning in aerospace. Section 2.2 considers prescriptive maintenance studies, where decision support is addressed more directly but is typically framed as planning or optimisation. Section 2.3 reviews semantic, knowledge-based, and modular orchestration approaches, which contribute enabling mechanisms for interoperability, coordination, and traceability. This progression supports the final subsection in identifying the unresolved gap addressed by the developed architecture.

2.1. Diagnostic, Prognostic, and Platform-Level IVHM Architectures

Early systems were designed to identify fault states and support structured troubleshooting. NASA's Livingstone is representative of this class (Williams & Nayak, 1996). Developed for spaceflight applications, it used component-based declarative models and a discrete propositional inference engine to compare predicted and observed behaviour and maintain candidate system hypotheses during operation (Sweet & Bajwa, 2003). Livingstone's propositional logic is form of rule-based reasoning, which it used to track planner goals, diagnose faults, and coordinate responses across hybrid hardware and software systems. Its dependence on a centralised inferential loop, however, generally required the system to settle into a quiescent state before diagnosis could be performed reliably (Hayden et al., 2005). This limited its suitability for the transient dynamics and asynchronous commanding typical of aircraft subsystems.

Research subsequently extended model-based reasoning into distributed settings. TEAMS-RDS (Testability Engineering and Maintenance System - Remote Diagnostic Server) adopted a three-tier architecture comprising sensor agents, a broker layer, and reasoning engines for remote fault isolation over a network (Deb & Ghoshal, 2001). It demonstrated remote monitoring and technical-manual integration; however, it remained reactive in function. The architecture centred on a master diagnostic engine that computed fault-isolation probabilities from predefined pass/fail test outcomes (Ferrell & Oostdyk, 2010). These systems established important principles for structured reasoning and distributed support. Nonetheless, they remained focused on fault identification rather than on coordinated, forward-looking maintenance action.

Recognising the limitations of component-isolated diagnostics, subsequent work moved toward higher-level aircraft platform-level reasoning. FAVER was introduced to address vehicle-level diagnosis by reasoning across interacting aircraft systems rather than treating subsystems independently (Ezhilarasu & Jennions, 2021). It utilised a virtual aircraft model combining DTs and reasoning techniques to isolate local faults, identify cascading effects across flight phases, and determine platform-level root causes. Its reported capability to trace an environmental-control-system problem to an originating engine fault illustrates the value of this approach. Subsequent platform health management review work has similarly argued that aircraft maintenance requires a platform-level treatment of fault interactions (Kabashkin, 2024). A related development, although not yet a DT orchestration architecture in the present sense, is the machine learning-based diagnostics and prognostics framework of Adhikari et al., which was developed to accelerate aircraft predictive-maintenance capability under limited run-to-failure data (Adhikari et al., 2019). The aforementioned framework was supported by a simulation environment linking onboard health assessment, offboard analytics, and enterprise-level maintenance and logistics planning, and was demonstrated on landing gear, electrical-system, and engine case studies. In this respect, it broadened the literature from isolated diagnosis toward generic predictive-maintenance infrastructure. However, both lines of work remain oriented toward fault reasoning, prognostic algorithm development, and decision-support benchmarking rather than toward prescriptive coordination. Beyond diagnostic and platform-level fault reasoning, later IVHM research increasingly incorporated predictive-maintenance functionality through hybrid DTs, prognostic modelling, and uncertainty-aware monitoring.

Recent aviation DT research has strengthened the prognostic side of IVHM: Norcaro et al. proposed a DT-based IVHM approach based on an authoritative hybrid as-operated twin that integrates physics-informed artificial intelligence, explicit verification and validation, and model-based systems engineering continuity between design and operational phases in order to enable predictive maintenance (Norcaro et al., 2025). The researchers positioned their work against conventional approaches that are limited by scarce run-to-failure data, uncertainty in system behaviour and prediction, and fragmentation between design and maintenance processes, and framed their contribution in terms of progression toward Type III IVHM capability. Related work on distributed aero-engine health monitoring addressed uncertainty-aware data prioritisation and synchronisation between onboard and off-board twins under severe computational and communication constraints (Hartwell et al., 2024). These studies improve predictive capability, model credibility, and deployment feasibility. They do not, however, specify how heterogeneous subsystem DTs would be selected and sequenced in response to a maintenance-related query.

2.2. Prescriptive Maintenance

Prescriptive maintenance in aerospace has been developed more explicitly in operations research and maintenance planning than in executable IVHM reasoning architectures. Prescriptive maintenance extends predictive maintenance by coupling anticipated failure with recommended maintenance action (Orošnjak et al., 2025). Meissner et al. addressed this directly through a discrete-event simulation framework for post-prognostic decision making in aviation (Meissner et al., 2021). Using the PreMaDe tool, they derived prescriptive maintenance strategies for an Airbus A320 tyre-pressure measurement task (Meissner et al., 2021). Their results showed that condition-monitoring information can be translated into maintenance decisions with measurable effects on service waiting times, maintenance scheduling, and fleet utilisation (Meissner et al., 2021). A related body of work by Koizumi and Kogiso considered aircraft health management from a system-of-systems perspective. They formulated health management as a multi-objective optimisation problem and modelled the strategic, economic, and logistical interdependencies among operators, maintenance facilities, and regulatory actors (Koizumi & Kogiso, 2025). Work in this area also integrated predictive models with stochastic programming to account for resource limitations, physical separation between facilities, and production gaps in complex equipment clusters (Mao et al., 2023). Crucially, this approach incorporates chance constraints; a probabilistic risk-management mechanism used to guarantee that the number of simultaneously unavailable or failed units remains below a specific threshold with a highly defined probability. By formally bounding the risk of concurrent equipment downtime amidst the high uncertainty of industrial production schedules, this shifted the literature from isolated, component-level prognostics toward stakeholder-coupled optimisation and real-world operational contexts.

These studies generally formulate prescriptive maintenance as strategic or offline planning rather than as an executable reasoning process embedded within an operational maintenance-support framework. Prognostic outputs are typically treated as static inputs to a separate planning stage, rather than being dynamically interpreted, combined, and acted upon within a unified query-driven process (Meissner et al., 2021). Recent review work on artificial intelligence, machine learning, and DT technologies in aircraft and broader transportation maintenance supports this distinction, showing that predictive maintenance, condition monitoring, and fault diagnosis remain dominant applications, while autonomous planning and advanced decision support are still emergent and unevenly adopted (Pavlyuk & Alomar, 2026; Werbińska-Wojciechowska et al., 2024). The limitation of this strand is therefore not the absence of optimisation or prescriptive intent, but the treatment of prognostic outputs as inputs to planning rather than as part of an executable reasoning process.

2.3. Semantic, Knowledge-Based, and Modular Orchestration Approaches

As DT ecosystems have expanded, a parallel research strand has focused on semantics, interoperability, and reasoning transparency. In regulated domains such as aviation, explainability is tied to certification and assurance requirements (Teubert et al., 2023). Ontology-driven DT frameworks have therefore been proposed to represent domain concepts in a machine-interpretable form. Kabashkin presented an ontology-driven DT framework for aviation maintenance and operations that integrates functional, behavioural, monitoring, lifecycle, and environmental ontologies into a unified semantic model (Kabashkin, 2025). The aforementioned model supports causal reasoning, rule-based validation, and context-aware maintenance recommendations, providing explainable and traceable decision support. These contributions are relevant as they show the importance of explicit reasoning objectives in aviation frameworks.

Architecturally relevant orchestration approaches have also been reported outside aviation maintenance. Su et al. proposed a DT system for manufacturing processes based on a multi-layer knowledge graph and demonstrated it for aero-engine blade manufacturing (Su et al., 2025). Their architecture linked a physical environment, a three-layer knowledge graph comprising Concept, Model, and Decision layers, and a virtual environment, with the knowledge graph serving as the interface for information exchange and process optimisation. Stavropoulou et al. likewise integrated a DT with a knowledge graph to orchestrate Virtual Objects in a glass-bending manufacturing environment (Stavropoulou et al., 2024). In the latter architecture, a composite Virtual Object acted as a master controller, polling the knowledge graph for device configurations and issuing commands to subordinate Virtual Objects. These studies demonstrate how semantic layers can support modular configuration rather than simple data storage. Their orchestration targets, however, are manufacturing workflows and process coordination rather than aircraft component utilisation reasoning. Beyond ontology and knowledge-graph-based approaches, the broader systems engineering literature demonstrates a shift toward modular coordination through agent-based, rule-based, and neuro-symbolic reasoning schemes (Ezhilarasu & Jennions, 2021; Hakim et al., 2025; Jiang & Hu, 2025; Kodors et al., 2025).

Within the predictive maintenance domain, Jiang and Hu presented an artificial intelligence agent-enabled framework comprising distinct diagnostic, scheduling, evaluation, and DT-based simulation agents (Jiang & Hu, 2025). By integrating multi-agent collaboration with retrieval-augmented knowledge, their architecture addressed maintenance scheduling under resource constraints. Expanding outside of aircraft-specific applications to examine the broader landscape of modular coordination, the RecGen framework offers a relevant, transparent cross-domain baseline. Designed as a no-coding shell for a rule-based expert system, RecGen integrates DT and Capability-Driven Development elements to provide decision support, validated specifically in the context of school food waste reduction rather than industrial maintenance (Kodors et al., 2025). Although its architecture provides a highly interpretable heuristic engine for a prescriptive DT, it fundamentally depends on static spreadsheet imports for its knowledge base and lacks native dynamic learning capabilities. While these deterministic constraints limit its direct suitability for the complex, high-frequency real-time demands of aviation prognostics and health management, RecGen remains a highly useful architectural example of how capability-driven reasoning can be structurally mapped to a DT interface. Returning to dynamic industrial applications, neuro-symbolic DT architectures such as ANSR-DT, developed by Hakim et al., attempt to bridge this gap between dynamic neural learning and symbolic rule transparency. However, from a rigorous systems design perspective, these hybrid frameworks continue to face critical scalability bottlenecks, particularly when continuous neural-network outputs must be translated into discrete symbolic rules with sufficient confidence and low enough latency to ensure reliable, real-time operational adaptation (Hakim et al., 2025).

2.4. Scope of Present Work

Taken together, this literature contributes three elements required by the present study: explicit knowledge representation, modular coordination, and traceable recommendation logic. The literature reviewed establishes that the principal technical elements required for prescriptive maintenance exist, albeit distributed across separate research strands, namely, diagnostic architectures, vehicle-level reasoning frameworks, predictive DTs, and semantic coordination models. Livingstone and TEAMS-RDS provided structured model-based fault isolation and troubleshooting (Deb & Ghoshal, 2001; Sweet & Bajwa, 2003). FAVER extended reasoning to interacting aircraft systems and cascading fault analysis (Ezhilarasu, 2019). Recent aviation DT work contributed hybrid predictive models, uncertainty-aware updating, and digital continuity (Norcaro et al., 2025). Prescriptive maintenance frameworks showed that health information can be converted into operational decisions with measurable effects on service timing and fleet utilisation (Meissner et al., 2021). Semantic and ontology-driven frameworks provided the mechanisms for interoperability, traceability, and explainable reasoning (Kabashkin, 2025; Su et al., 2025).

What remains insufficiently addressed is the integration of these capabilities at the level of component remaining useful life reasoning. Specifically, the literature provides limited evidence of an aircraft-oriented architecture that can accept a maintenance query, determine which heterogeneous subsystem DTs and engineering algorithms are required, coordinate their execution across coupled subsystems without imposing a fixed upstream–downstream dependency hierarchy, and collate their outputs into a prescriptive recommendation with explicit traceability and uncertainty representation. Existing work addresses parts of this problem, but not their coordinated execution within a unified query-driven architecture.

3. DEVELOPED HIGH-LEVEL REASONER ARCHITECTURE

The High-Level Reasoning Layer (HLR), shown in the grey box in Figure 1, is proposed as a supervisory reasoning layer for prescriptive maintenance. It operates as a ground-based core positioned between the subsystem DT layer (green box in Figure 1) and the user-facing decision interface (questions menu to the left of the HLR in Figure 1). This aligns with IVHM guidelines that distribute functions across subsystem, platform, and enterprise levels rather than confining reasoning to a single component (SAE HM-1, 2019, 2023a).

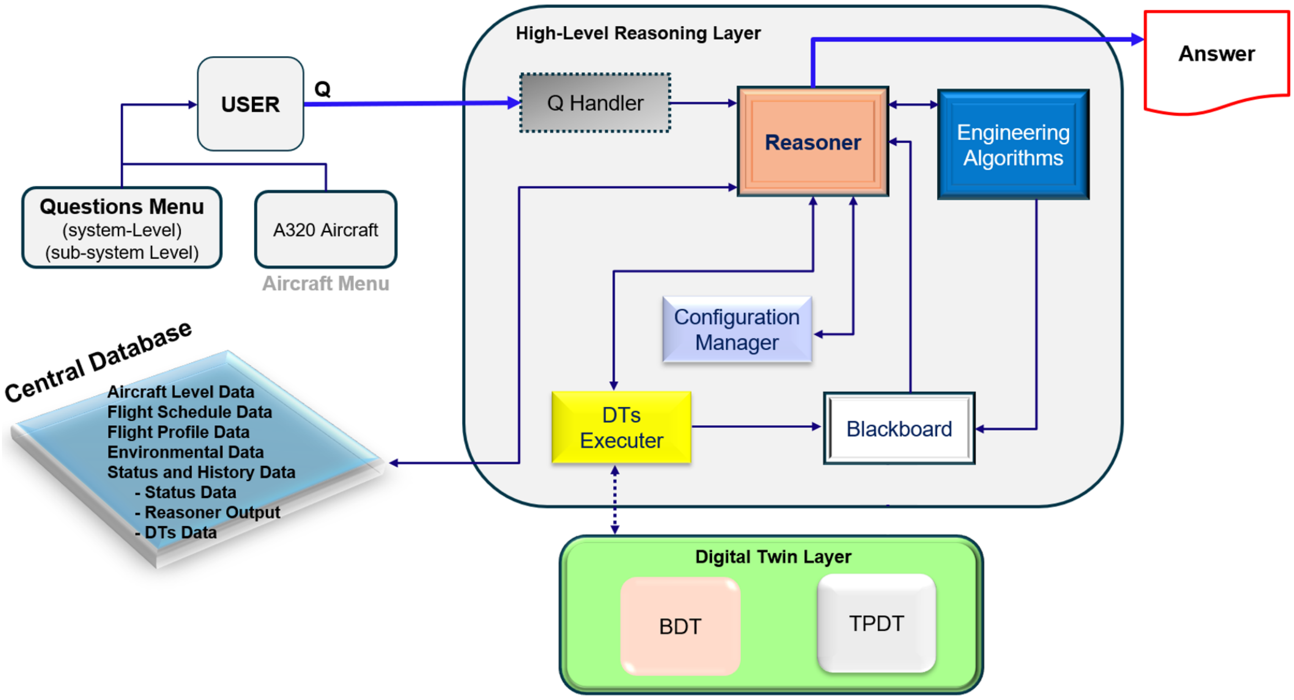

Figure 1 Architecture View of the HLR, highlighting inter-module data flow and direction.

3.1. Design Rationale and Architectural Principles

Aircraft component usage and life reasoning requires more than the isolated execution of a prognostic model: it necessitates integrating subsystem states, operational history, environmental context, and maintenance constraints. Building on vehicle-level reasoning frameworks such as FAVER (Ezhilarasu, 2019) and positioned alongside recent aviation DT assurance work concerned with hybrid model credibility, Validation, Verification, and Accreditation (VVA), and digital continuity (Norcaro et al., 2025), the HLR acts as a meta-reasoning layer that extends platform-level fault interpretation toward query-driven coordination of subsystem DTs, engineering algorithms, and maintenance evidence for prescriptive decision support.

The architectural formulation of the HLR is informed by established aerospace industry standards and recommended practices, which collectively provide the necessary scaffolding for complex IVHM systems. Documents such as SAE JA6268 (SAE HM-1, 2023), ARP6290 (SAE HM-1, 2023b), ARP6407 (SAE HM-1, 2019), and ARP6887 (SAE HM-1, 2024) are foundational to this domain; they establish standardised taxonomies, functional allocation boundaries across enterprise and subsystem levels, and assured approaches to system design, interoperability, and validation. By adhering to these guidelines, the architecture ensures regulatory alignment and a structured methodology for decomposing complex platforms into manageable, health-ready components. However, these standards inherently serve as high-level systems engineering frameworks rather than prescriptive software implementations. While they successfully define what functional capabilities an IVHM architecture must possess, where they should be allocated, and how they must be verified, they do not dictate the specific computational orchestration, dynamic memory structures, or exact algorithmic fusion mechanisms required to execute prescriptive queries at runtime. Consequently, the HLR architecture is designed to bridge this gap, translating these conceptual guidelines into an executable, data-driven orchestration plane capable of active reasoning. The standards provide the systems-engineering framework, whereas the HLR provides the executable architecture that realises runtime orchestration and reasoning.

The first design principle is prescribed functional modularity. Each architectural unit is assigned a bounded responsibility, while cross-module interactions are mediated through the HLR Reasoner module rather than through direct pairwise dependencies among DTs and algorithms. This reduces integration burden and supports incremental extension of the architecture. A comparable modular strategy was demonstrated in FAVER, which was shown to adapt to additional aircraft systems, including the engine, without altering the wider reasoning structure (Ezhilarasu & Jennions, 2021).

The second design principle is separation between query semantics and model implementation. In practice, different stakeholders ask different questions: one user may ask for remaining useful life (RUL), and another for the reason behind abnormal coupled degradation. The architecture as a result does not bind a question directly to a fixed subsystem model. Instead, the question is first interpreted as a query type, after which it is then mapped by the Reasoner module to the required DTs, engineering algorithms (where applicable), and order of module execution. This is consistent with the SAE Recommended Practice documentation ARP6883, which treats concept of operations, use cases, interfaces, and data inventory as upstream drivers of IVHM system requirements (SAE HM-1, 2019), and with ARP6290 and ARP6887, which place operational use cases at the centre of architecture definition and end-to-end validation (SAE, 2010; SAE HM-1, 2024).

The third principle is system-of-systems compatibility. ARP6887 explicitly characterises IVHM as a system of systems distributed across airborne, ground, and enterprise elements, with intended function achieved only through coordinated operation of all constituent parts (SAE HM-1, 2024). That principle is directly relevant here. Aircraft comprise multiple interacting systems with different sensing provisions, degradation mechanisms, update rates, and ownership boundaries. A prescriptive architecture shall not assume identical data structures, identical uncertainty representations, or identical computational characteristics across these systems. It must instead provide a mechanism for controlled heterogeneity.

The fourth principle is functional layering with nonlocal decision support. ISO 13374 and relevant IVHM guidance organise health functions into data acquisition, data manipulation, state detection, health assessment, prognostic assessment, and advisory generation (ISO/TC 108/SC 5, 2003; SAE HM-1, 2023). ARP6407 further notes that advisory generation is often performed nonlocally since actionable advice frequently depends on correlation across subsystems and enterprise information (SAE HM-1, 2019). The HLR adopts this principle explicitly: lower-level DTs remain responsible for subsystem-specific estimation and prediction, while the HLR performs the cross-subsystem evaluation and prescriptive reasoning needed to transform model outputs into maintenance advice.

For the aforementioned reasons, the developed HLR is developed as a modular, off-board architectural layer whose purpose is to accept component life relevant queries, identify the appropriate expert modules, coordinate their execution, preserve reasoning traceability, and return prescriptive recommendations with explicit confidence. As shown in Figure 1, the architecture separates orchestration functions from data-processing functions and structures query execution through distinct sensor (Central Database), DT-processing (DT Layer), and prescriptive advisory (Answer) layers. It is important to note that the landing gear use case in this study is an instantiation of the architecture rather than its limit of applicability.

3.2. Core Components and Functionalities

The HLR comprises seven principal architectural elements, allocated in accordance with the functional layering defined in ISO 13374 (ISO/TC 108/SC 5, 2003), as well as the architectural tiering principles established in SAE ARP6290 (SAE HM-1, 2023b). ARP6290 is a foundational aerospace recommended practice that provides the specific framework for translating high-level operational goals into a structured, multi-tiered IVHM architecture, ensuring that data capture, transfer, and analysis are appropriately bounded. Consequently, the HLR elements are intentionally defined with prescribed functions rather than overlapping responsibilities. This bounded allocation of responsibility is what allows the HLR to be updated or expanded by modifying local modules, interfaces, or metadata, without requiring redesign of the full reasoning chain. As shown in Figure 1, these elements are distributed between a control/orchestration plane (the HLR Layer) and a layered data-processing structure (the DT Layer and the Central Database). Together, these elements define the HLR as an architecture with fixed component responsibilities and interaction pathways; its framework-like generalisability arises from the modularity of those interfaces rather than from abstraction alone. Their primary responsibilities are detailed in Table 1.

The Question Handler module is the entry point for maintenance-relevant part availability and status intent. Its purpose is to transform a user-facing request into an executable internal representation. In the present architecture, this is achieved through structured query templates. The Question Handler identifies the query class. This function reflects the standards emphasis on use cases and stakeholder goals as the basis for IVHM system behaviour (SAE HM-1, 2023b, 2024).

The Configuration Manager module governs discoverability and applicability of expert modules. Its purpose is to hold the metadata that allow the architecture to determine which DTs and algorithms are relevant to a given query, what their input prerequisites are, what validity constraints apply, and whether their outputs can be combined. This information is outlaid to the Reasoner module. This is necessary as IVHM components differ in both inherent functionality and transparency. This requirement aligns with SAE JA6268 (SAE HM-1, 2023), a critical standard designed to reduce system integration barriers by specifying how supplier-provided components must expose their capabilities. JA6268 supports this by requiring suppliers and integrators to expose enough design-time and run-time semantics for higher-level systems to interpret condition indicators, health indicators, prognostic conclusions, and maintenance workplans (SAE HM-1, 2023). In the HLR, these semantics are recorded as configuration metadata rather than left implicit in code.

The Reasoner module is the orchestration core. It does not itself replace the subsystem DTs; rather, it sequences their implementation, which is invoked via the DTs executer module, invokes supporting Engineering Algorithms where necessary, and applies the reasoning strategy appropriate to the query. This approach is aligned with SAE ARP6407, which provides the overarching systems engineering framework for IVHM design across component, subsystem, and enterprise levels. ARP6407 notes that platform-level reasoning depends on the knowledge captured in fault models, the level of abstraction chosen, and the type of runtime data available, and that the reasoning or aggregation function may be approached from model-based, rule-based, statistical, or case-based perspectives (SAE HM-1, 2019). The HLR follows the same principle in architectural terms. In the present implementation, deductive logic is used where hard constraints or policy limits apply, causal reasoning where subsystem interactions must be interpreted, and statistical reasoning where uncertainty-bearing predictions must be fused. This provides the operational basis for query-driven orchestration within the landing gear use case.

The Blackboard module is the shared working memory of the architecture. Its function is to store intermediate states, which include derived quantities from the Engineering Algorithms block and DTs executer outputs while a query is being resolved. This component is necessary since prescriptive queries frequently require staged analysis rather than immediate one-pass inference. One subsystem may need to execute first in order to provide inputs or context to another, and each subsystem must preserve both intermediate and final states, given explainability is required. In the HLR, the Blackboard therefore serves both operational and assurance purposes: operationally, it permits incremental multi-module fusion; from an assurance standpoint, it preserves the lineage needed for auditability and later validation, fulfilling the stringent traceability and state-preservation requirements mandated by the validation and verification (V&V) frameworks in ARP6887 (SAE HM-1, 2024). ARP6887 is essential for defining the systematic V&V processes required for IVHM certification.

The DTs Executer module facilitates data transfer within the DTs Layer, which serves as the execution environment for subsystem-specific predictive and diagnostic models, explicitly encapsulating the Health Assessment (HA) and Prognostic Assessment (PA) decision-support blocks of the ISO 13374 functional reference model. Each DT operates as an expert module with defined inputs, outputs, applicability limits, and execution constraints, as governed by the Configuration Manager. This deliberate partitioning reflects SAE ARP6407's mandate that IVHM design requires the strategic allocation of functions across subsystem, platform, and enterprise tiers, rather than assuming a homogenous distribution of the ISO 13374 blocks throughout the architecture (SAE HM-1, 2019). Furthermore, this abstraction strictly aligns with the SAE JA6268 specification for health-ready components. Under this standard, lower-level components and their local controllers are tasked with implementing the foundational Data Acquisition (DA), Data Manipulation (DM), and State Detection (SD) functions directly at the edge (SAE HM-1, 2023). By securely exposing sufficient design-time and run-time data, these health-ready components enable the upstream DT Layer to cleanly execute computationally intensive reasoning tasks, specifically; Health Assessment (HA) to determine the current state of equipment health, and Prognostic Assessment (PA) to forecast future health states and performance life remaining, without violating strict software architectural boundaries.

The Engineering Algorithms module facilitates derived health calculations following subsequent DT runs facilitated by the DTs executer, with the intermediary Reasoner module facilitating DTs to Engineering Algorithms data exchange. In the present architecture, this block is responsible for computing prognostic metrics such as RUL and other advisory variables derived from DT outputs, thresholds, and query context. As shown in Figure 1, the Engineering Algorithms block exchanges data bidirectionally with the Reasoner and writes its outputs to the Blackboard for subsequent use by the Reasoner. This separation preserves modularity by distinguishing subsystem-level model execution from cross-model derived calculation.

The Central Database provides the persistent information required to make prescriptive reasoning possible. In the current implementation, it stores aircraft-level operational history, subsystem telemetry, flight-segment information, environmental data, maintenance history and component usage per flight cycle, DT outputs, and historical HLR outputs, with output saves facilitated through the Reasoner module. This is consistent with ARP6407 recommendations, which note that enterprise-level IVHM may require historical data for trending, fault alerts for isolation, maintenance history for validation, and data collection for life-cycle cost analysis (SAE HM-1, 2019). In the present landing gear instantiation, the database stores FlightRadar24-derived operational records, together with estimated landing weights derived from initial take-off-weight approximations and ICAO fuel-burn rates for the selected engine, and ISA-table estimates of runway ambient temperature. The status and history portion of the database also provides the basis for retaining historical and newly accumulated cases, thereby supporting future extension of the reasoning architecture toward case-based maintenance recommendation.

Table 1 Core modules of the HLR architecture

Module	**Operational requirement**	**Task**
Question Handler	Different stakeholders/MRO personnel ask different questions.	Maps user queries to predefined templates.
Configuration Manager	DTs and expert modules have different availability, validity domains, and prerequisites.	defines DT and algorithms data dependencies.
Reasoner	Multiple reasoning strategies and expert modules must be coordinated.	Sequences execution of DTs and Engineering Algorithms, applies reasoning strategies, constructs query results, facilitates data transfer between modules.
Blackboard	Intermediate results and derived quantities must remain visible and traceable during execution.	Stores intermediate Engineering Algorithm and DTs executer outputs

Module	**Operational requirement**	**Task**
Digital Twins Layer	Subsystems have different physics, sensing, and data availability.	Encapsulated DTs (e.g., BDT, TPDT) with standardised data-transfer contracts, validity domains, and configuration models.
Engineering Algorithms	Derived quantities such as RUL require cross-model calculation and query-specific evaluation.	Computes RUL and other derived advisory variables from DT outputs, thresholds, and context data from the Central Database.
Central Database	Reasoning requires aircraft status, operational history, schedules, environment, and prior DT runs.	Stores aircraft state, flight profiles, sensor data, environmental data, maintenance context, DTs executer outputs and historical HLR outputs.

3.3. Multi-Layered Data Flow Architecture

While the physical instantiation follows the four-tier IVHM architecture defined in ARP6290; comprising the On-vehicle, Data transfer, At-vehicle, and Fleet data services tiers (SAE HM-1, 2023b), the HLR's logical data flow consists of three distinct transfer layers: sensor data, DT-processed features, and postprocessed prescriptive outputs. Separating these layers prevents the conflation of data capture, model execution, RUL and uncertainty calculations, and decision-making.

The sensor data layer: represented by the Central Database, handles operational evidence, including time-series measurements and flight-context data, acquired post flight. Operating to serve the ground-based reasoner rather than an airborne engine, this layer corresponds to ISO 13374 data acquisition and manipulation (ISO/TC 108/SC 5, 2003). ARP6290 notes that data acquisition generally occurs in the on-vehicle tier, while downstream processing may be allocated off-board when timeliness constraints are lower, storage and transmission are feasible, or raw data capture is required for development and validation (SAE HM-1, 2023b).

The DT-processed feature layer contains transformed subsystem outputs and derived algorithmic quantities. It includes the subsystem DTs and their exchanged outputs. Following JA6268 principles (SAE HM-1, 2023), each DT must expose machine-usable condition and health indicators that can be consumed by subsequent modules without human reinterpretation. The DT outputs and algorithm-derived quantities are written to the Blackboard as a common intermediate evidence state.

To support end-to-end scenario validation (SAE HM-1, 2024), the data flow is designed for strict traceability. The Blackboard preserves intermediate states, Engineering Algorithm outputs, and data dependencies, ensuring the architecture remains auditable and verifiable across both its physical tiers and logical functional blocks.

3.4. Extensibility Mechanisms

A central architectural requirement, following the modular design principle introduced in Section 3.1, is extensibility. The HLR is intended to support the progressive addition of new subsystem DTs without redesign of the orchestration core. This requirement follows directly from both the aircraft context and the standards context. ARP6290 cautions that IVHM architectures must accommodate heterogeneous data sources, external interfaces, and incremental integration (SAE HM-1, 2023b). ARP6407 similarly treats integrated IVHM design as an iterative process in which features, interfaces, triggers, algorithms, and data collection schemes are progressively refined as new needs and constraints emerge (SAE HM-1, 2019). The HLR therefore implements extensibility through a registration-based interface contract rather than through hard-coded module coupling.

For a new DT or Engineering Algorithm module to become schedulable by the HLR, it must declare four elements. First, it must provide an I/O (data transfer) specification describing required inputs and produced outputs. Second, it must define its validity domain, including operating conditions, aircraft applicability, and any assumptions under which its outputs remain trustworthy. Third, it must provide a statement of computational characteristics, since orchestration requires awareness of execution cost and sequencing constraints. Fourth, it must declare any dependencies on other modules or prior processing stages. These principles are compatible with JA6268, which distinguishes between run-time interface semantics and design-time data needed for higher-level integration, and which explicitly recommends machine-interpretable formats that maximise automated interoperability between subsystem suppliers and higher-level systems Design & Run-Time Information Exchange for Health-Ready Components, 2023).

Once registered, the Configuration Manager reveals the DT or Engineering Algorithm module to the Reasoner as an available expert module, requiring no modification to the core orchestration logic. For example, a new shock-strut DT may be integrated by registering its required inputs and outputs, while a new advisory algorithm may be introduced by declaring the DT outputs and thresholds on which it depends. This metadata-driven registration prepares the architecture for future semantic integration, allowing the architecture to scale from fixed proof-of-concept models to broader aircraft-system reasoning environments. This scalability gives the architecture framework-like generalisability, but the primary contribution remains the architecture itself.

3.5. Landing Gear Use Case and Digital Twin Instantiation

The landing gear system serves as an ideal demonstration domain due to the tight thermo-mechanical coupling between its subsystems. During ground operations, brake thermal loading directly drives tyre temperature and pressure trends, which in turn constrains the braking operational envelope. Since the RUL of these components is governed by this interacting degradation rather than independent trajectories, single-subsystem predictions are insufficient for prescriptive queries such as life extension or root-cause analysis. Consequently, the HLR is required to perform dependency-aware orchestration and causal reasoning across the subsystem models rather than merely aggregating independent outputs.

The current instantiation contains two landing-gear DTs: the Brake Temperature Digital Twin (BDT) and the Tyre Pressure Digital Twin (TPDT). Test flights containing temporal brake temperature and application data, as well as tyre pressure data, were used to tune the models to the available aircraft data and to provide limited validation before the DTs were applied to operational records derived from FlightRadar24. For the FlightRadar24 records, inputs not directly available in the flight data were approximated using initial take-off-weight estimates, ICAO fuel-burn rates for the selected engine, and ISA tables for on-ground runway ambient temperature. These derived variables provided the flight context needed by the BDT and TPDT within the HLR workflow.

3.5.1. Brake Temperature Digital Twin (BDT)

The Brake Temperature Digital Twin (BDT) is a data-driven model that estimates brake thermal response from operational telemetry. A data-driven formulation was selected to ensure the low computational cost required for repeated, query-driven execution without the overhead of a high-fidelity thermo-mechanical solver. Operating gate-to-gate, the BDT segments missions into characteristic phases (take-off, descent, touchdown, taxi, and parking) using curve-fitting techniques. It outputs brake temperature, which serves as a leading indicator for subsequent tyre pressure modelling.

3.5.2. Tyre Pressure Digital Twin (TPDT)

Similar to the BDT, the Tyre Pressure Digital Twin (TPDT) predicts tyre pressure using fitted relationships informed by operational, environmental, and thermal inputs. These inputs include ISA-derived ambient conditions, estimated landing weight, and the brake thermal outputs generated by the BDT. Within the HLR, this dependency is managed through the Configuration Manager, which provides the Reasoner with an order of execution where the DT Executer ensures that the required BDT outputs are available to the TPDT via the Blackboard during query execution. This allows the TPDT to estimate tyre pressure in a modular manner without hard-coded links.

3.5.3. DT Interface Standardisation

To integrate these heterogeneous models, both the BDT and TPDT conform to the standardised interface contract established in Section 3.4, requiring explicit declarations of I/O, validity domains, computational costs, and module dependencies. This metadata-driven standardisation allows the HLR Reasoner and Configuration Manager to validate query executability and sequence the DTs. DT coordination is therefore handled through explicit model contracts rather than hard-coded coupling, preserving the modularity of the orchestration core while execution and data transfer are handled by the DTs Executer.

4. REASONING STRATEGIES AND QUERY PROCESSING

4.1. Multi-Strategy Reasoning Scheme

Following the architectural definition and landing gear instantiation in Chapter 3, prescriptive maintenance requires transitioning from isolated fault detection to actionable decision support. Single-strategy approaches, relying exclusively on machine learning predictions or static rule bases, are insufficient to handle the diversity of maintenance queries. The HLR therefore implements a multi-strategy reasoning scheme, selecting the appropriate logic according to the query context:

- Deductive (Rule-Based) Reasoning: applied when policy compliance or hard operational limits dictate maintenance constraints (e.g., dispatch criteria).
- Statistical (Prognostic) Reasoning: utilised to manage degradation under uncertainty, quantifying confidence bounds for RUL estimates.
- Causal Reasoning: required to interpret subsystem interactions and evaluate how degradation in one component (e.g., brake thermal loading) accelerates wear in another (e.g., tyre pressure loss).
- Abductive Reasoning, with future extension to case-based reasoning: deployed to resolve conflicting predictions across DTs and, in future implementations, to draw on historical operational when formulating maintenance alternatives.

4.2. Example Queries and End-to-End Orchestration

The HLR architecture is fundamentally query-driven; the user's operational intent dictates the execution sequence, the specific DTs invoked, and the reasoning strategy applied. Table 2 outlines three standardised query templates demonstrating this end-to-end orchestration within the landing gear use case.

Table 2 Example queries and corresponding HLR orchestration

Query	**Represents**	**DTs Required**	**Reasoning Strategies**	**Expected Output to User**
Q1: Brake Temperature Prediction Profile	"What is the brake temperature profile for a flight in 2 days?"	BDT	Data-driven prediction, temporal profile estimation	Forecast brake-temperature profile, peak temperature
Q2: Tyre Pressure Prediction	"What is the tyre pressure for a flight in 2 days?"	BDT, TPDT	Causal chaining, physics-informed prediction	Forecast tyre pressure, pressure trajectory
Q3: Brake/Tyre Replacement	"When will the tyre/brake require replacement?"	BDT, TPDT	Prognostic reasoning, causal dependency, deductive threshold evaluation	Predicted replacement point, limiting subsystem, confidence / uncertainty drivers from Eng. Alg.

4.3. Departure from Diagnostic Workflows

This query-driven architecture represents a fundamental shift from conventional diagnostic workflows to prescriptive optimisation. Traditional diagnostic systems (such as FAVER or TEAMS-RDS) focus on retrospective fault isolation. A diagnostic output typically concludes: "Aircraft N has excessive brake wear; the likely root cause is debris ingestion or hydraulic failure."

In contrast, the HLR's prescriptive approach synthesises operational context to determine future action, quantifying the trade-offs of different maintenance interventions. A prescriptive output example may conclude: "Aircraft N exhibits accelerated brake wear driven by high landing-weight profiles and short post-touchdown high-speed taxiing durations. Prescriptive action: Increasing reverse thrust by 20% or upgrading brake material will extend tyre life by 100 hours. Proceed based on current MRO constraints." By quantifying these trade-offs, the HLR shifts the architectural objective from simply identifying what went wrong to defensibly recommending what to do next.

While herein the HLR was instantiated within the landing gear domain, its metadata-driven, model-agnostic design ensures scalability across aircraft subsystems and transferability to other safety-critical industries. The core architectural gap addressed by this research; the industry-wide transition from isolated predictive models to coordinated prescriptive orchestration, is equally prevalent in sectors managing complex, interacting cyber-physical systems.

5. SUMMARY AND CONCLUSIONS

Existing IVHM and DT frameworks excel at isolated prediction. However, they lack the multi-model orchestration mechanisms required to answer prescriptive maintenance queries with step-by-step auditability. To bridge this gap, this paper has developed the High-Level Reasoner, a modular meta-reasoning innovative architecture that advances maintenance from predictive to prescriptive. The HLR achieves this through an structured orchestration pipeline: a Query Handler interprets operational intent, a Configuration Manager dynamically discovers and tasks appropriate expert modules, a Blackboard provides a transparent workspace for data agglomeration, a DTs executer handles data transfer and operation of the specialised independent DTs, and a hybrid Reasoner applies deductive, causal, and statistical logic, with the architecture designed to support future extension toward case-based reasoning through its retained status and history data.

The developed architecture has been demonstrated on an aircraft landing gear system (ATA 32), successfully coordinating heterogeneous brake and tyre DTs. By quantifying subsystems dependencies, the HLR synthesised predictive outputs into defensible prescriptive recommendations, such as RUL maximisation and causal root-cause analysis, which exceed the capabilities of conventional diagnostics and prognostics systems. By facilitating a shift from reactive scheduling to optimisation-driven strategies, this architecture supports the aviation industry's objective of reducing MRO costs without compromising safety or operational availability. Its modular design also provides a reusable framework for extension across additional aircraft subsystems and future IVHM applications.

6. FUTURE WORK

Future work may examine the implementation challenges encountered during development, particularly those associated with interface definition, module integration, orchestration logic, and the iterative refinement of the Reasoner. This would help clarify the practical issues involved in progressing from the present proof-of-concept architecture toward a more mature implementation.

ACKNOWLEDGEMENT

This research was funded by Innovate UK grant number 10002411, under the ATI/IUK Project: LANDOne, with Airbus UK as Industrial Lead.

REFERENCES

Adhikari, P., Rao, H. G., & Buderath, Dipl.-I. M. (2019). Machine Learning based Data Driven Diagnostics & Prognostics Framework for Aircraft Predictive Maintenance. *E-Journal of Nondestructive Testing*, *24*(03). https://www.ndt.net/search/docs.php3?id=23875

Anandavel, S., Li, W., Garg, A., & Gao, L. (2021). Application of digital twins to the product lifecycle management of battery packs of electric vehicles. *IET Collaborative Intelligent Manufacturing*, *3*(4), 356–366. https://doi.org/10.1049/cim2.12028

Bastard, G., Lacaille, J., Coupard, J., & Stouky, Y. (2016). Engine Health Management in Safran Aircraft Engines. *Annual Conference of the PHM Society*, *8*(1), 101–108. https://doi.org/10.36001/phmconf.2016.v8i1.2523

Bernard, W., & Hoffmann, A. (2023). SKYWISE - Big data platform as a foundation of airlines predictive and health monitoring. *PHM Society Asia-Pacific Conference*, *4*(1). https://doi.org/10.36001/phmap.2023.v4i1.3722

Brandt, S. (2017). Monitoring, understanding and managing engine data. *Maintenance Cost Conference*. https://www.iata.org/contentassets/af577ffae2714e859202927be50afbea/1400-1430-monitoring-understanding-managing-eng-data-mtu.pdf

Butter, M. (2019). *Event-based Risk Quantification in Flight Data Analysis*. Technical University of Munich.

Condition Monitoring and Diagnostics of Machines. Data Processing, Communication and Presentation - General Guidelines, Pub. L. BS ISO 13374-1:2003 (2003). https://knowledge.bsigroup.com/products/condition-monitoring-and-diagnostics-of-machines-data-processing-communication-and-presentation-general-guidelines

Corrêa Macedo, F., César de Andrade Silva, B., & Ghiorzi Varela Parente, E. (2025). Analysis of maintenance planning for PW1100G aircraft engines and the implications for operational safety. *Aeronautics and Aerospace Open Access Journal*, *9*(3). https://doi.org/10.15406/aaoaj.2025.09.00234

Deb, S., & Ghoshal, S. (2001). Remote diagnosis server architecture. *AUTOTESTCON (Proceedings)*, 988–998. https://doi.org/10.1109/autest.2001.949478

Design & Run-Time Information Exchange for Health-Ready Components, Pub. L. JA6268_202303 (2023). https://doi.org/10.4271/JA6268_202303

El Mir, H., King, S., Skote, M., & Perinpanayagam, S. (2023). *Machine learning requirements for the airworthiness of structural health monitoring systems in aircraft*. International Committee on Aeronautical Fatigue (ICAF). https://www.icaf.aero/icaf2023/proceedings/documents/128.pdf

Elliott, J. (2021, October). The IntelligentEngine. *Aerospace Manufacturing and Design*. https://www.aerospacemanufacturinganddesign.com/article/the-intelligentengine/

Ezhilarasu, C. M. (2019). *A framework for aerospace vehicle reasoning (FAVER)* [Cranfield University]. https://dspace.lib.cranfield.ac.uk/handle/1826/20305

Ezhilarasu, C. M., & Jennions, I. K. (2021). Development and Implementation of a Framework for Aerospace Vehicle Reasoning (FAVER). *IEEE Access*, *9*, 108028–108048. https://doi.org/10.1109/ACCESS.2021.3100865

Ferrell, B., & Oostdyk, R. (2010, March 6). Modeling and Performance Considerations for Automated Fault Isolation in Complex Systems. *2010 IEEE Aerospace Conference*. https://ntrs.nasa.gov/citations/20100036468

Forest, F. (2021). Unsupervised Learning of Data Representations and Cluster Structures : Applications to Large-scale Health Monitoring of Turbofan Aircraft Engines [Université Sorbonne Paris Nord]. In *Focus on Microscopy* (Number 1). https://doi.org/10.34894/VQ1DJA

Guidelines for Development of Civil Aircraft and Systems, Pub. L. ARP4754B, 400 Commonwealth Drive, Warrendale, PA, United States: SAE International (2023). https://doi.org/10.4271/ARP4754B

Guidelines for the Development of Architectures for Integrated Vehicle Health Management Systems, Pub. L. ARP6290 (2023). https://doi.org/10.4271/ARP6290

Guidelines for the Development of Architectures for Integrated Vehicle Health Management Systems, Pub. L. ARP6290 (2023). https://doi.org/10.4271/ARP6290

Hakim, S. Bin, Adil, M., Velasquez, A., & Song, H. H. (2025). *ANSR-DT: An Adaptive Neuro-Symbolic Learning and Reasoning Framework for Digital Twins*. http://arxiv.org/abs/2501.08561

Hartwell, A., Montana, F., Jacobs, W., Kadirkamanathan, V., Ameri, N., & Mills, A. R. (2024). Distributed digital twins for health monitoring: resource constrained aero-engine fleet management. *The Aeronautical Journal*, *128*(1325), 1556–1575. https://doi.org/10.1017/aer.2024.23

Hayden, S. C., Sweet, A. J., & Shulman, S. (2005). Lessons learned in the livingstone 2 on earth observing one flight experiment. *Collection of Technical Papers - InfoTech at Aerospace: Advancing Contemporary Aerospace Technologies and Their Integration*, *2*, 863–877. https://doi.org/10.2514/6.2005-7000

Health management System for new 777-300ERs. (2006). *Aircraft Engineering and Aerospace Technology*, *78*(3). https://doi.org/10.1108/aeat.2006.12778cab.024

IVHM Design Guidelines, Pub. L. ARP6407 (2019). https://doi.org/10.4271/ARP6407

Jiang, W., & Hu, F. (2025). Artificial Intelligence Agent-Enabled Predictive Maintenance: Conceptual Proposal and Basic Framework. *Computers 2025, Vol. 14, Page 329*, *14*(8), 329. https://doi.org/10.3390/computers14080329

Kabashkin, I. (2024). Digital Twin Framework for Aircraft Lifecycle Management Based on Data-Driven Models. *Mathematics 2024, Vol. 12, Page 2979*, *12*(19), 2979. https://doi.org/10.3390/math12192979

Kabashkin, I. (2025). Ontology-Driven Digital Twin Framework for Aviation Maintenance and Operations. *Mathematics 2025, Vol. 13, Page 2817*, *13*(17), 2817. https://doi.org/10.3390/math13172817

Kodors, S., Apeinans, I., Zarembo, I., & Lonska, J. (2025). RecGen: No-Coding Shell of Rule-Based Expert System with Digital Twin and Capability-Driven Approach Elements for Building Recommendation Systems. *Applied Sciences (Switzerland)*, *15*(19), 10482. https://doi.org/10.3390/app151910482

Koizumi, T., & Kogiso, N. (2025). A System of Systems Architecture for Optimizing Aircraft Health Management in Civil Aviation. *PHM Society Asia-Pacific Conference*, *5*(1). https://doi.org/10.36001/phmap.2025.v5i1.4423

Mao, H., Liu, Z., Qiu, C., Huang, Y., & Tan, J. (2023). Prescriptive maintenance for complex products with digital twin considering production planning and resource constraints. *Measurement Science and Technology*, *34*(12), 125903. https://doi.org/10.1088/1361-6501/aced5f

Meissner, R., Rahn, A., & Wicke, K. (2021). Developing prescriptive maintenance strategies in the aviation industry based on a discrete-event simulation framework for post-prognostics decision making. *Reliability Engineering and System Safety*, *214*. https://doi.org/10.1016/j.ress.2021.107812

Norcaro, S., Cumbo, R., Petrella, V., Mangeruca, L., Di Guglielmo, L., & Ulisse, A. (2025). Digital Twin-based IVHM for Predictive Maintenance. *Annual Conference of the PHM Society*, *17*(1). https://doi.org/10.36001/phmconf.2025.v17i1.4343

Orošnjak, M., Saretzky, F., & Kedziora, S. (2025). Prescriptive Maintenance: A Systematic Literature Review and Exploratory Meta-Synthesis. *Applied Sciences (Switzerland)*, *15*(15), 8507. https://doi.org/10.3390/app15158507

Pavlyuk, D., & Alomar, I. (2026). Artificial Intelligence Technologies for Aircraft Maintenance. *International Journal of Prognostics and Health Management*, *17*(1). https://doi.org/10.36001/ijphm.2026.v17i1.4567

Prentice, B., DiNota, A., Dulac, N., Hayes, L., Ferguson, I., Holden, D., Mishra, U., & Huang, R. (2023). *Global Fleet And MRO Market Forecast 2024-2034: Key Trends*. https://www.oliverwyman.com/our-expertise/insights/2024/feb/global-fleet-and-mro-market-forecast-2024-2034.html

Shukla, B., Fan, I.-S., & Jennions, I. (2020). Opportunities for Explainable Artificial Intelligence in Aerospace Predictive Maintenance. *PHM Society European Conference*, *5*(1), 11–11. https://doi.org/10.36001/phme.2020.v5i1.1231

Skaltsis, G. M., Wang, C., Plastropoulos, A., Fan, I. S., & Avdelidis, N. P. (2025). Wireless data transfer system architecture for predictive maintenance of aircraft landing gear. *AIAA Science and Technology Forum and Exposition, AIAA SciTech Forum 2025*. https://doi.org/10.2514/6.2025-1659

Stavropoulou, G., Tsitseklis, K., Mavraidi, L., Chang, K. I., Zafeiropoulos, A., Karyotis, V., & Papavassiliou, S. (2024). Digital Twin Meets Knowledge Graph for Intelligent Manufacturing Processes. *Sensors 2024, Vol. 24, Page 2618*, *24*(8), 2618. https://doi.org/10.3390/s24082618

Su, C., Tang, X., Jiang, Q., Han, Y., Wang, T., & Jiang, D. (2025). Digital twin system for manufacturing processes based on a multi-layer knowledge graph model. *Scientific Reports*, *15*(1), 12835. https://doi.org/10.1038/s41598-024-85053-0

Sweet, A., & Bajwa, A. (2003, December 1). Lessons Learned from using a Livingstone Model to Diagnose a Main Propulsion System. *JANNAF 39th CS/27th APS/21st PSHS/3rd MSS Joint Subcommittee Meeting*. https://ntrs.nasa.gov/citations/20040045151

Teubert, C., Pohya, A. A., & Gorospe, G. (2023). *An Analysis of Barriers Preventing the Widespread Adoption of Predictive and Prescriptive Maintenance in Aviation*. https://ntrs.nasa.gov/citations/20230000841

Validation and Verification of Integrated Vehicle Health Management Systems, Pub. L. ARP6887 (2024). https://doi.org/10.4271/ARP6887

Werbińska-Wojciechowska, S., Giel, R., & Winiarska, K. (2024). Digital Twin Approach for Operation and Maintenance of Transportation System—Systematic Review. *Sensors*, *24*(18), 6069. https://doi.org/10.3390/s24186069

Williams, B. C., & Nayak, P. P. (1996). A Model-Based Approach to Reactive Self-Configuring Systems. *Proceedings of the AAAI Conference on Artificial Intelligence*, *13*. https://aaai.org/papers/144-aaai96-144-a-model-based-approach-to-reactive-self-configuring-systems/