Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 140 (2026) 677–682

www.elsevier.com/locate/procedia

33rd CIRP Conference on Life Cycle Engineering (LCE 2026)

Predictive Maintenance for Life Cycle Engineering Using I4.0 Technologies in MRO Data Systems

Marco Weiss[a,*], Florian Raddatz[a], Gerko Wende[a]

[a]German Aerospace Center, Institute of Maintenance, Repair and Overhaul, Hein-Sass-Weg 22, Hamburg, 21129, Germany

* Corresponding author. *E-mail address:* Marco.Weiss@DLR.de

Abstract

Efficient Life Cycle Engineering relies on credible insight into how assets degrade in operation. Predictive maintenance has been identified as a key enabler, but its data management is often fragmented across heterogeneous, proprietary systems and does not readily feed back into design, maintenance planning, or end-of-life decisions. This paper presents and experimentally verifies an Industry 4.0-based approach that embeds predictive maintenance into a standardized digital-twin architecture, centered on the Asset Administration Shell, to close this gap. Firstly, to standardize data and enable intelligent processing, we upgrade our existing Industry 4.0 Component Stack to predict and manage key condition indicators, such as remaining useful life, together with event logic, and to employ our proposed Digital Product Passport that records thresholds, events, and maintenance outcomes in interoperable submodels. Secondly, to support context-aware decision making, we show how the upgraded stack maps event codes derived from cross-sensor telemetry to decision guards that trigger service tasks such as operation shaping or unplanned maintenance and is designed for continuous review and learning throughout the life cycle. The approach is implemented and verified on an unmanned aircraft system used as a surrogate asset in a maintenance, repair, and overhaul data environment, demonstrating that assets can locally interpret their operating context, autonomously request appropriate interventions, and expose the resulting evidence through standardized interfaces. For life cycle engineers, the approach transforms degradation behavior, interventions, and operation outcomes from isolated runtime logs into traceable, machine-readable information that directly supports design choices, maintenance strategies, and end-of-life planning.

Keywords: Industry 4.0; Circularity; Predictive Maintenance; Decision-Making; Asset Administration Shell; Digital Product Passport; Data Space

1. Introduction and Background

1.1. *Life Cycle Engineering and Predictive Maintenance*

The transition towards sustainable, cost-efficient, and resilient asset management is increasingly embedded in the principles of Life Cycle Engineering (LCE), which emphasize operational continuity, waste reduction, and circular economy alignment. Within this context, the concept of Predictive Maintenance (PdM) has become a key enabler by providing dynamic insights into asset health and Remaining Useful Life (RUL). PdM contributes directly to LCE goals by reducing unplanned downtimes, extending component lifespans, and lowering maintenance costs. PdM has demonstrated substantial impact on industrial efficiency: as described in [1], it can reduce scheduled repairs by up to 12 % and overall maintenance costs by 30 %, while predicting around 70 % of potential failures. In addition, recent artificial intelligence driven PdM models achieved a prediction accuracy of 92 %, resulting in a 35 % reduction in downtime and a 28 % decrease in maintenance costs, outperforming traditional maintenance approaches [2]. Taken together, these results illustrate how PdM transforms maintenance from a reactive cost center into a strategic driver of efficiency and sustainability.

10.1016/j.procir.2026.05.114

Nomenclature

AAS	Asset Administration Shell
ADS	Asset Data Server
CC	Constant Current
CV	Constant Voltage
DC	Dataspace Connector
DPP	Digital Product Passport
EFC	Equivalent Full Cycles
FSM	Finite State Machine
I4.0CS	Industry 4.0 Component Stack
IDTA	Industrial Digital Twin Association
IIoT	Industrial Internet of Things
MRO	Maintenance, Repair and Overhaul
OCV	Open-Circuit Voltage
OPC UA	Open Platform Communications Unified Architecture
PdM	Predictive Maintenance
REST	Representational State Transfer (Interface)
RUL	Remaining Useful Life
SoC	State of Charge
SM	Submodel
TSD	Time Series Data
UAS	Unmanned Aircraft System

1.2. *Asset Administration Shell as Enabler of PdM*

In Industry 4.0 (I4.0), the Asset Administration Shell (AAS) provides a syntactic and semantic framework required to scale predictive maintenance [3, 4]. By encapsulating operational data, health indicators, and life cycle information in machine-interpretable submodels, the AAS enables interoperability and integration of predictive algorithms [5, 6]. AAS-based "digital twins" facilitate dynamic condition monitoring and RUL estimation, but their early Type 1 and Type 2 implementations remain limited to structured data provision and reactive monitoring [7]. Proactive AAS has been introduced, embedding autonomous logics and negotiation capabilities that allow assets to initiate actions and interact with peers without centralized orchestration [8]. Demonstrations in Maintenance, Repair, and Overhaul (MRO) [9, 10] show proactive AAS instances monitoring thresholds, negotiating services, and exchanging condition data within cyber-physical production systems. Recent work also integrates AI/ML models via Industrial Digital Twin Association (IDTA) submodels [11], providing a pathway to embed predictive models in an I4.0 framework.

1.3. *Challenges and Research Gap*

Despite the demonstrated benefits, data-driven predictive maintenance still faces obstacles in industrial deployment. A major issue is heterogeneity of assets, sensors, and protocols, which complicates unified monitoring and limits interoperability despite existing semantic submodels [7]. In addition, Industrial Internet of Things (IIoT) environments generate non-stationary and noisy time series [12], undermining the reliability and generalizability of detection models developed in controlled settings [13].

Validation under realistic operating conditions also remains insufficient. As the authors of [14] highlight, validation of predictive approaches often remains confined to simulations or limited pilots, lacking systematic evaluation under realistic, heterogeneous, yet controllable conditions. Likewise, [15] observes that PdM methods seldom undergo large-scale, safety-critical validation because real-world failures are rare and operational systems are complex. Finally, the alignment of PdM with digital twin architectures continues to face persistent gaps in explainability, scalability, and standardization, as highlighted in recent systematic reviews [16]. Complementing this, [17] proposes a requirement-based roadmap, emphasizing the definition of clear functional and information requirements as essential to enable standardized, industry-scale automation.

These challenges motivate PdM frameworks that are interoperable, resilient to data drift, and validated in realistic operation. This study evaluates selected I4.0 technologies on an Unmanned Aircraft System (UAS) surrogate to enable scalable, autonomous PdM and support life cycle optimization in heterogeneous data ecosystems.

2. Architectural Framework

2.1. *Industry 4.0 Component Stack*

To contribute to dynamic life cycle management, we upgrade our Industry 4.0 Component Stack (I4.0CS) [9, 10, 18, 19], which enables real-world assets to participate in digital environments. As illustrated in Figure 1, this stack transforms raw asset data into life cycle-relevant information and makes it available to other stakeholders via federated data spaces. The asset may be physical or logical; combined with its AAS it forms an "I4.0 Component." Data originating from sensors, logs, or monitoring functions – accessible through controllers, APIs, or IIoT gateways – constitutes the foundation for digital processing and representation. Together with the asset, the I4.0CS consists of the following primary modules:

The **Asset Data Server** (ADS) provides an abstraction layer between the asset and its digital shell. It acquires data from physical and logical interfaces, preprocesses it (filtering, unit conversion, basic analysis), and exposes structured information via standardized protocols such as OPC UA or REST. This digital interface creates semantic visibility of the asset and serves as the southbound data source for the AAS server.

The **AAS server** is implemented as a Type 3 AAS with an integrated architecture that exchanges data with assets and maps it into submodels, thereby operationalizing the AAS metamodel [3] in a runtime system. In our framework [9, 18], this server is composed of modules that manage the in-memory AAS data system, operate an internal event bus, and handle inbound and outbound communication. These modules embed finite state machines (FSM) and rule engines, which meet the requirements of autonomy, intelligence, and collaboration [8]. In practice, this enables asset-specific logic for decision-making, indicator evaluation, service negotiation, and process orchestration. Standardized interfaces, including a REST API and protocols defined by the I4.0 language [20] ensure interoperability with other I4.0-compliant entities. While functionally comparable to other frameworks such as BaSyx and FA³ST, the distinguishing feature is the use of the proactive capabilities of the Type 3 AAS. Recent research highlights the complementarity of Multi-Agent Systems (MAS) and AAS, with MAS providing distributed intelligence to strengthen

Type 3 implementations [8]. In this view, each proactive AAS server can be regarded as an autonomous agent representing its asset, engaging in peer-to-peer communication to coordinate maintenance or production tasks. Integrating (or tightly coupling) agent logic preserves standardized structures while enabling adaptive, proactive life cycle behavior.

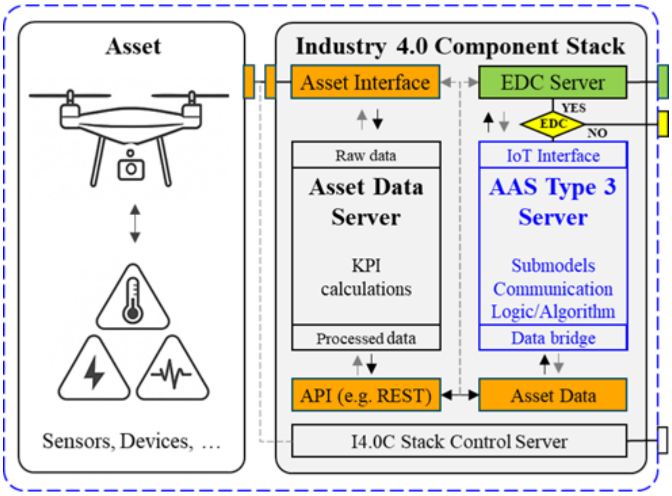

Fig. 1. (a) Asset; (b) Industry 4.0 Component Stack

The I4.0CS uses a **dataspace connector** to enable secure, policy-controlled exchange in federated data ecosystems. Based on the same AAS-driven information model and communication semantics introduced above, assets act as independent participants, sharing life cycle and compliance information with maintenance providers, operators, designers, and evaluators under full traceability.

2.2. *Specific Adaptations for Predictive Maintenance*

In the I4.0CS, predictive maintenance capabilities are not treated as standalone add-ons but as integral parts of the asset's digital life cycle representation. Building on our earlier work [19], we implement an I4.0-compliant Digital Product Passport (DPP, Figure 2) on the AAS metamodel [3] to ensure seamless interoperability. The DPP is operationalized on the AAS Server as a modular construct comprising IDTA submodels (e.g., [21]) and domain-specific extensions such as Battery and Electric Drive Passports. As illustrated in Figure 2, these modules enable component-level exchange while maintaining overall DPP continuity – e.g., replacing a battery automatically updates its associated passport. This design enforces semantic and syntactic consistency, life cycle traceability, and compliance with emerging regulations that require passport-based reporting for batteries and other components.

Within this DPP, the Predictive Maintenance (PdM) submodel [22] holds the related key condition indicators (KCI) such as Remaining Useful Life (RUL), associated confidence intervals, and pre-alert thresholds, while the complementary Time Series Data (TSD) submodel [23] records the evolution of these indicators over the asset's lifetime. Both submodels are referenced via embedded sub-passports, such as the "BatteryPass" in Figure 2. This ensures that the condition and degradation histories of a component are unambiguously linked to its unique identity, usage record, and compliance information, even when the component is replaced.

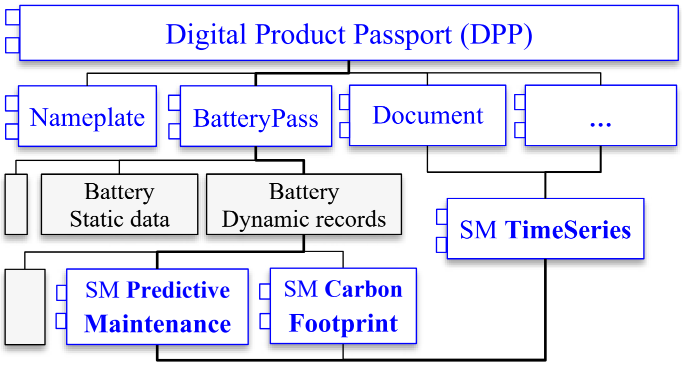

Fig. 2. Digital Product Passport

To detail the specific adaptations, we choose the battery as the power-supply subsystem because it exhibits measurable degradation behavior and provides well-structured telemetry suitable for predictive maintenance evaluation. The Asset Data Server publishes voltage, current, temperature, and state-of-charge (SoC) from UAS sensors and the flight controller, to which the AAS server subscribes. Per-mission charge-discharge behavior is then captured automatically by recording three SoC values – before operation (SoC_1), after operation (SoC_2), and after recharge (SoC_3) – from which the equivalent partial $\Delta Cycle$ is computed as

$$\Delta Cycle = \frac{1}{2}\left(\frac{(SoC_1 - SoC_2)}{100} + \frac{(SoC_3 - SoC_2)}{100}\right) \tag{1}$$

with SoC_x in [%]. This formulation normalizes full, partial, and incomplete recharge scenarios, providing a consistent basis for usage-driven degradation. Summing these increments yields equivalent full life cycles (EFC; 1.0 means one full charge-discharge equivalent). After each mission n, two health observables are calculated: capacity-based state of health (SoH_n) (0,1] and normalized internal resistance R_n/R_0 with which the effective lifetime L_{eff} is estimated in an empirical combination: The remaining useful lifetime per cycle RUL_n is then given by the difference between the effective L_{eff} and cumulated battery cycles as

$$RUL_n = max\left(0, \overbrace{\frac{L_0 SoH_n^{\gamma(z_n)}}{1+\beta(z_n)\cdot\left(\frac{R_n}{R_0}-1\right)_+}}^{L_{eff}} - \underbrace{\sum_{t=1}^{n} \Delta Cycle(t)}_{EFC_n}\right) \tag{2}$$

where L_0 is the rated lifetime [cycles], $(x)_+ = \max(x, 0)$ enforces resistance penalties only when $R_n > R_0$ and z_n are measured stress covariates (e.g., C-rate, depth of discharge, temperature, cool-down). The sensitivities β [1] and γ [1] are learned from data and updated as more cycles are observed.

Figure 3 illustrates the related event-driven, context-aware logic: The AAS server subscribes to the Asset Data Server's telemetry (SoC, voltage, current, vibration, throttle settings, and ambient conditions) and updates the PdM submodel with new RUL predictions and the TSD submodel with the chronology of the related SoC, ΔCycle, and RUL. For this, it runs three concurrent mechanisms: (i) *anomaly monitoring*, (ii) *recharge detection*, and (iii) *operation-mode monitoring*:

The *anomaly monitoring* (i) continuously evaluates critical variables with thresholds and hysteresis; on violation, PreAlerts are written to the PdM submodel, mirrored to the TSD submodel and triggers other events, thus as service requests like an unplanned battery replacement.

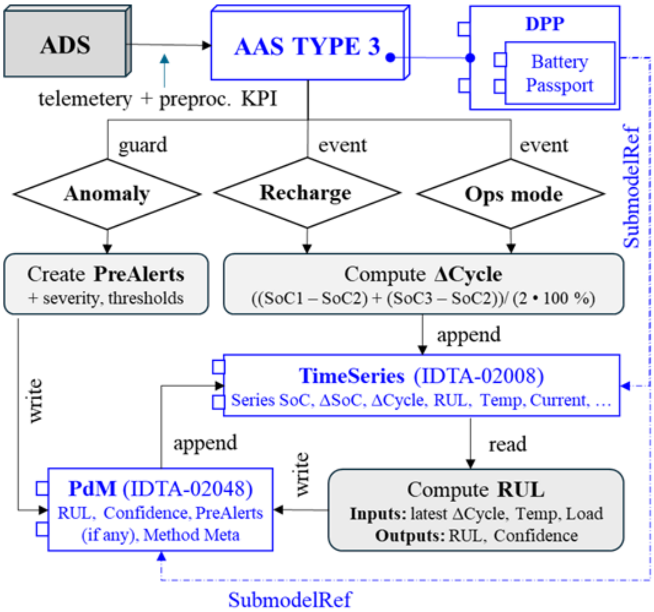

Fig. 3. Data flow in the PdM architecture of the I4.0CS

The *recharge detection* (ii) operates on events: the system evaluates a steady increase in SoC, the sign and magnitude of the charge current, the Constant Current–Constant Voltage (CC–CV) charging profile, and the plausibility of the SoC-Open-Circuit-Voltage (SoC-OCV) relation. At charge end, ΔCycle is finalized from the SoC triplet and appended – together with ΔSoC – to the TSD. The AAS persists the last valid SoC at battery disconnect so that, when a recharged battery is reconnected, degradation continuity is preserved.

The *operation-mode monitoring* (iii) tracks state transitions (e.g., power off → on → off) based on controller signals or derived indications (throttle, airspeed, vibration). A segment-end (use) event sets (SoC_1, SoC_2); the cycle is finalized only after the subsequent recharge provides SoC_3.

3. Implementation and Operation

Building directly on the framework introduced in Section 2, we demonstrate the utilization of the upgraded Industry 4.0 Component Stack (I4.0CS) on an Unmanned Aircraft System (UAS). The proactive AAS receives real sensor streams from the Asset Data Server (ADS) and uses its modularized finite state machine (FSM) architecture to process them and manage the results. This operationalizes the earlier-introduced closed loop: each cycle updates the power data records, produces a new remaining lifetime estimate and, if thresholds are violated, autonomously triggers alerts and maintenance activities.

3.1. *Architecture and runtime behavior*

The UAS exposes battery telemetry, including state of charge (SoC), pack voltage, current and temperature, via the flight controller and ADS. However, there are additional channels that feed the learning mechanism with data beyond the battery itself. This includes information such as the vehicle's vibration, attitude, ambient pressure and temperature, and strain gauges on the battery pack for monitoring abnormal swelling. Only by considering these cross-impact effects can we assess the system health comprehensively. The AAS processes these data and supervises their evolution in a LifeCycleMonitor FSM that operationalizes the prediction data flow from Section 2.2 (Figure 3). For isolation and provenance, the state machine runs in its own thread with five states, as illustrated in the transition diagram (Figure 4).

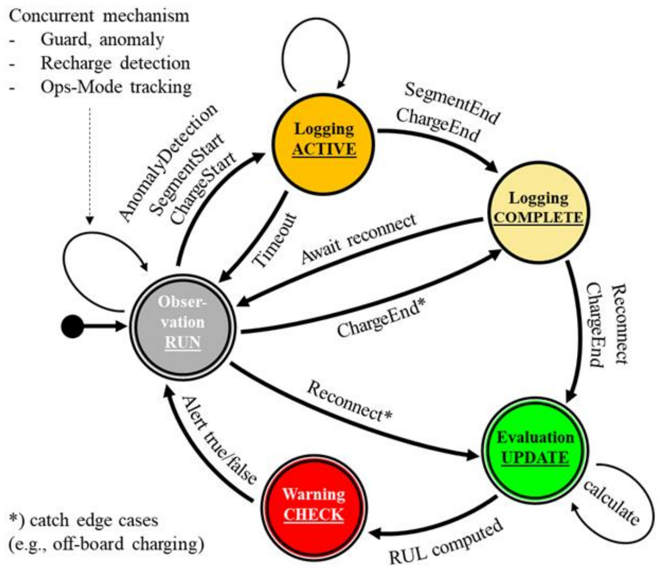

Fig. 4. AAS LifeCycleMonitor finite state machine transition diagram

In the main *Observation* state, the FSM instantiates three lightweight workers that execute the three concurrent mechanisms as introduced in Section 2.2: (i) anomaly monitoring, (ii) recharge detection, and (iii) operation-mode monitoring. The main thread remains authoritative: it performs nominal time-series recording, consumes worker events, and – on events ChargeStart, SegmentStart, or AnomalyDetected – switches to *LoggingActive*, raising the sampling rate and writing the buffered pre-event telemetry to the Time Series Data (TSD) submodel. The *LoggingComplete* state persists $SoC_{1,2}$ and, after events ChargeEnd or Reconnect, SoC_3; it then computes ΔCycle, and appends the SoC triplet and segment statistics to the Time Series Data submodel. In the Evaluation state, the FSM reads ΔCycle and cumulative features (e.g. cycles, temperature, vibration), estimates RUL (Eq. 2) and confidence, and writes them to the Digital Product Passport. Finally, WarningCheck evaluates threshold conditions and issues the corresponding PreAlerts.

4. Application and discussion

We verify the framework on an Unmanned Aircraft System (UAS) surrogate that could be substituted by other assets. Rather than optimizing the UAS itself, we show how the I4.0CS manages multi-sensor interdependencies and adapts operational strategies. While many studies emphasize learning a degradation model, we focus on the benefits of overall system integration: The I4.0CS captures, prepares and supervises the data from the UAS, including signals beyond its power system, such as propulsion proxies (e.g. RPM), vibration, attitude, ambient conditions and mission context. It then translates this information into key condition indicators (KCI), such as remaining useful life (RUL) margins, mission duration trends and environmental impact intensity. These indicators are evaluated jointly (taking into account cross-sensor correlations and short-term trends), and then converted into advisories by performing an event-to-task mapping.

To ensure methodological consistency, we reused the experimental setup from our complementary study [10], and extended it with a computer-controlled power supply capable of emulating various battery behaviors, including deep or light discharge and degradation profiles. In this setup, the UAS and power supply are manipulated to simulate different fault conditions. As discussed in [10], there is a cross-impact between propulsion anomalies and power supply behavior, e.g. in terms of changed consumption and thus discharge, affecting the battery health in turn, for the same task. In this paper, we add an advanced prediction model using the RUL formulation introduced in Section 2.2, where mission-specific stress covariates are used to adapt the lifetime estimate.

4.1. *Cross-correlated lifetime and impact indicators*

Figure 5 presents a simulated mission in which all indicator values are normalized. The objective is to verify runtime behavior – such as detection latency, trend fitting, event stamping, and task execution – rather than to report asset-specific absolute lifetime values. After an initial normal segment, a mechanical imbalance anomaly is introduced and intensified via a throttle ramp-up.

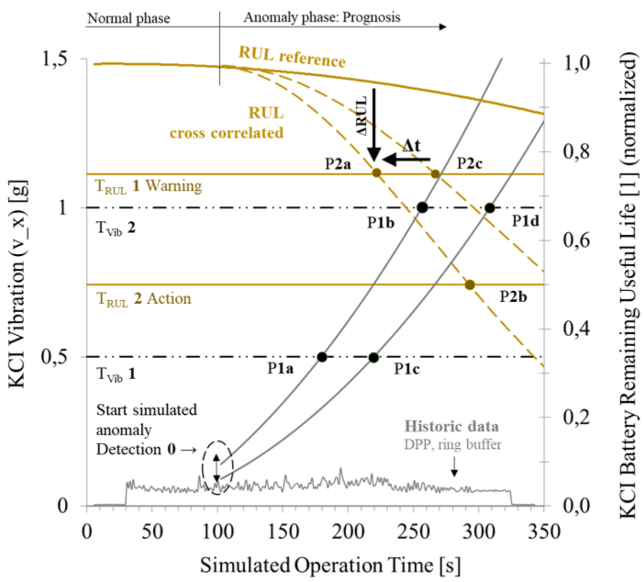

Fig. 5. Prediction of early (points a-b) and late (c-d) events

This results in two solid-lined vibration projections that rise from Detection 0 and cross the thresholds $T_{Vib}1$ and $T_{Vib}2$ at points P1a–P1b (early warning) and P1c–P1d (late warning), respectively. In parallel, the I4.0CS computes three remaining useful life (RUL) projections based on the data-driven calibration of Formula (2). These represent the conservative power-supply-sensors-only baseline ("RUL normative") and the advanced integrated multi-sensor curves ("RUL cross-correlated"), where the lifetime model in Formula (2) also incorporates mechanical-sensor data. They fall from Detection 0 and cross the thresholds $T_{RUL}1$ and $T_{RUL}2$ at points P2a–P2b and P2c–P2d. Early and late warnings are derived from the statistical bandwidth. As the throttle ramp-up intensifies the coupled electro-mechanical state, the multi-sensor curve reaches the warning and action thresholds earlier (P2a / P2c) and stays lower within the prognosis window compared to the normative RUL curve. These results show that, compared to the normal single-sensor method, the I4.0 multi-sensor approach offers two major benefits: it detects anomalies sooner (Δt), and it adjusts the RUL downward more accurately in response to coupled electro-mechanical stress (ΔRUL), thereby improving prognostic reliability.

4.2. *From indicators to events, tasks, and life cycle decisions*

The behavior shown in Section 4.1 verifies that the developed logic reacts as intended, but on its own it is not sufficient to optimize life cycle strategies. To interpret joint patterns as life cycle-relevant guards, these are mapped to a predefined event matrix, which assigns combinations of threshold violations across electrical and mechanical indicators to classify event codes (Table 1). For example, when vibration exceeds $T_{Vib}2$ and battery health in terms of RUL drops below $T_{RUL}1$ but remains above $T_{RUL}2$, the observation is classified as event code #5.

Table 1. Event-code classification matrix

Indicator Vibration	Indicator Battery RUL	Event Code
$> T_{Vib}$ 1 and $< T_{Vib}$ 2	$> T_{RUL}$ 1	#1
$> T_{Vib}$ 2	$> T_{RUL}$ 1	#2
$< T_{Vib}$ 1	$< T_{RUL}$ 1 and $> T_{RUL}$ 2	#3
$> T_{Vib}$ 1 and $< T_{Vib}$ 2	$< T_{RUL}$ 1 and $> T_{RUL}$ 2	#4
$> T_{Vib}$ 2	$< T_{RUL}$ 1 and $> T_{RUL}$ 2	#5
$< T_{Vib}$ 1	$< T_{RUL}$ 2	#6
$> T_{Vib}$ 1 and $< T_{Vib}$ 2	$< T_{RUL}$ 2	#7
$> T_{Vib}$ 2	$< T_{RUL}$ 2	#8

Table 2. Event code → decision guards / task mapping (reduced)

Event Code	Type	Assigned tasks
#1	moderate mechanical issue	• Log anomaly and tag affected missions as "monitor". • Request unplanned propulsion inspection in the next possible maintenance window.
…	…	…
#4	moderate mechanical & power issue	• Apply operation shaping. • Initiate unscheduled maintenance. • Advise battery replacement.
…	…	…
#8	severe end-of-life	• Restrict immediately to low-criticality use. • Initiate immediate maintenance actions.

These event codes are then mapped to decision guards that trigger tasks such as condition-based inspections, mission shaping, scheduled or unscheduled pack replacements, or adjustments of charging policies. Table 2 exemplifies how these codes translate into concrete tasks that are then requested automatically as services. For example, event #1 triggers enhanced monitoring and planned inspection, event #4 activates mission shaping and accelerated maintenance, and event #8 leads to reclassification, usage restriction, and preparation for immediate battery replacement.

This event-task mapping forms an important intermediate layer between numeric thresholds and high-level decision guards. It establishes a structured scheme in which indicators, thresholds, and event assignments are continuously reviewed and revised throughout the asset's life cycle, thereby supporting ongoing optimization. In [10, 19], we introduced the complementary module of the proactive Asset Administration Shell, where an EventManager state machine continuously monitors these indicators, assigns an event code when a defined pattern occurs (Table 1), and requests the corresponding tasks (Table 2). By executing the mechanism locally, each asset not only self-manages planned and unplanned interventions with minimal delay and without a central coordinator, but also generates consistent life cycle evidence on degradation and maintenance actions that can be directly exploited by LCE.

5. Conclusion

Life Cycle Engineering aims to base design, operation, and end-of-life decisions on how assets actually degrade in service. Today, this knowledge is still scattered across heterogeneous, proprietary data sources that are hard to compare over time, across assets, or between organizations. The approach presented in this paper goes beyond simply closing this gap: it turns predictive maintenance into an integral part of a standardized, interoperable system architecture. Assets are no longer passive data sources, but active participants that interpret their own telemetry, classify events, and request suitable maintenance services on a shared semantic basis.

Within this architecture, condition indicators, remaining lifetime estimates, thresholds, and event codes are represented in standardized digital-twin structures and recorded in a Digital Product Passport together with all interventions and outcomes. Cross-functional signals from the overall system state are evaluated, improving health prediction and enabling adaptive decisions that are both explainable and auditable. Standardized communication and dataspace integration allow assets, operators, and maintainers to exchange information and negotiate actions without losing semantic consistency. For life cycle engineers, this means that degradation behavior, interventions, and mission outcomes become traceable, machine-readable evidence that directly informs design choices, maintenance planning, and end-of-life strategies, instead of remaining hidden in isolated runtime logs.

Acknowledgments

The authors gratefully acknowledge the support from The German Federal Ministry for Economic Affairs and Energy through the Aerospace-X project (Grant No.13MX004B) as part of the Manufacturing-X initiatives.

References

[1] M. Panagiotis, E. A. und D. Bechtsis, „Predictive maintenance in Industry 4.0: A systematic multi-sector mapping," *CIRP Journal of Manufacturing Science and Technology,* Bd. 50, pp. 80-103, 2024.

[2] R. A. Abdulrazzq, N. M. Sajid und M. S. Hasan, „Artificial intelligence-driven predictive maintenance in IoT systems," *South Florida Journal of Development,* Bd. 5, Nr. 12, 2024.

[3] DIN EN IEC 63278, *DIN EN IEC 63278-1 VDE 0810-781:2022-07 Asset Administration Shell for industrial applications,* Bd. VDE Standards, Berlin: VDE Verlag.

[4] IDTA, „Specification of the Asset Administration Shell Part 1: Metamodel," Industrial Digital Twin Organization, Frankfurt am Main, 2023a.

[5] J. Rahal, A. Schwarz, B. Sahelices, R. Weis und S. Antón, „The asset administration shell as enabler for predictive maintenance: a review," *Journal of Intelligent Manufacturing,* Bd. 2025, Nr. 1, pp. 19-33, 2023.

[6] R. Kerkeni, S. Khlif, A. Mhalla und K. Bouzrara, „Digital Twin Applied to Predictive Maintenance for Industry 4.0," *Journal of Nondestructive Evaluation, Diagnostics and Prognostics of Engineering Systems,* 2024.

[7] J. Zhang, C. Ellwein und M. Heithoff, „Digital twin and the asset administration shell," *Software and Systems Modeling,* Bd. 24, pp. 771-793, 8 January 2025.

[8] L. Sakurada, F. De la Prieta und P. Leitao, „The Role of Multi-Agent Systems in Realizing Asset Administration Shell Type 3," *Future Internet,* Bd. 17, Nr. 7, 20 June 2025.

[9] M. Weiss, H. Pakala, K. Wicke, M. Gill und G. Wende, „MaSiMO - Development and Research of Industry 4.0 Components with a Focus on Experimental Applications of Proactive Asset Administration Shells in Data-Driven Maintenance Environments," Stuttgart, 2023.

[10] M. Weiss, A. Kamtsiuris, F. Raddatz und G. Wende, „Advanced MRO Processes in Industry 4.0 with proactive Asset Administration Shell and Digital Product Passport," *International Journal of Prognostics and Health Management,* 2025.

[11] J. G. Marques, F. L. Medeiros, P. L. F. F. de Medeiros, G. B. P. Leitão, D. C. de Souza, D. R. C. Silva und L. A. Guedes, „Integrating Machine Learning into Asset Administration Shell: A Practical Example Using Industrial Control Valves," *Processes,* Bd. 13, Nr. 7, 2 July 2025.

[12] S. Fatima und A. Rahimi, „A Review of Time-Series Forecasting Algorithms for Industrial Manufacturing Systems," *Machines,* 2024.

[13] B. Kim, M. Alawami, E. Kim, S. Oh, J. Park und H. Kim, „A Comparative Study of Time Series Anomaly Detection Models for Industrial Control Systems," *Sensors,* 2023.

[14] O. O. Lawal, N. O. Nawari und O. Lawal, „AI-Enabled Cognitive Predictive Maintenance of Urban Assets Using City Information Modeling - Systematic Review," *Buildings,* Bd. 15, Nr. 5, 2025.

[15] T. Tinga, „Practical Application of Predictive Maintenance - Solving the Challenges," in *Proceedings of the 35th European Safety and Reliability Conference & the 33rd Society for Risk Analysis Europe Conference*, Stavanger, 2025.

[16] N. Abd Wahab, K. Hasikin, K. Wee Lai, K. Xia, L. Bei, K. Huang und X. Wu, „Systematic review of predictive maintenance and digital twin technologies challenges, opportunities, and best practices," *PeerJ Computer Science,* pp. 1-50, 22 April 2024.

[17] S. Ma, K. A. Flanigan und M. Bergés, „State-of-the-art review and synthesis: A requirement-based roadmap for standardized predictive maintenance automation using digital twin technologies," *Advanced Engineering Informatics,* Bd. 62, 2024.

[18] M. Weiss, K. Wicke und G. Wende, „MaSiMO - A Hybrid Experimental Platform for the Simulation and Evaluation of Data-Driven Maintenance Enterprises," in *DLRK2022 - NETZPUBLIKATIONEN*, Dresden, 2022.

[19] M. Weiss, F. Raddatz und G. Wende, „MaSiMO – Digital Product Passport and Autonomous Event Management of Industry 4.0 Components with Proactive AAS in Data-Driven Aviation Maintenance and Production," in *Deutscher Luft- und Raumfahrtkongress 2024*, Hamburg, 2024.

[20] VDI2193-1, *VDI/VDE 2193 Part 1:2020-04 Language for I4.0 components. Structure of messages,* Düsseldorf: VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik, 2020.

[21] IDTA, „Carbon Footprint," Industrial Digital Twin Association, Frankfurt am Main, 2025.

[22] IDTA, „Predictive Maintenance," Industrial Digital Twin Association, Frankfurt am Main, 2025.

[23] IDTA, „Time Series Data," Industrial Digital Twin Association, Frankfurt am Main, 2023.