

DIGITAL TWIN FOR AIR DEFENSE: LIVE-VIRTUAL-CONSTRUCTIVE INTEGRATION AND DECISION SUPPORT

Shalom Malka, Matan Ben Simon & Oleg Yakimenko

Naval Postgraduate School, Monterey, CA

Abstract

Integrated Air and Missile defence (IAMD) is a time-critical system of systems (SoS) comprising command-and-control systems, sensors, networks, and interceptors that must operate coherently under uncertainty. While digital twin (DT) concepts are increasingly adopted in aerospace and defence, their value for IAMD lies not primarily in high-fidelity physics modelling, but in the integration of synchronized operational state, executable models, and traceable decision-support workflows across the system lifecycle. This paper proposes a reference architecture for an IAMD Digital Twin explicitly designed to support Live-Virtual-Constructive (LVC) integration, Model-Based Systems Engineering traceability, and uncertainty-aware decision support. The architecture decomposes the DT into coordinated data, model, and decision layers, connected through a controlled digital thread that enables reuse of models, scenarios, and evidence across engineering, testing, training, operations, and sustainment. The architecture is instantiated through a Python-based operational research demonstrator initialized from a synchronized IAMD SoS state snapshot and evaluated using Monte Carlo simulation. Four representative trade-off studies are performed: engagement-policy selection, acquisition alternatives, fix-or-fight readiness decisions, and design sensitivity analysis. These studies are not intended to yield universal tactical or procurement conclusions; rather, they demonstrate how the proposed DT workflow links synchronized system state, executable assumptions, uncertainty treatment, and traceable decision products in a reproducible manner. The results provide initial proof of concept that an LVC-integrated DT architecture can support consistent, explainable, and scenario-specific decision support for complex IAMD SoS problems.

Keywords: integrated air and missile defence, digital twin, live-virtual-constructive integration, model-based systems engineering, decision support

1. Introduction and Background

Integrated Air and Missile defence (IAMD) is a complex physical system of systems (SoS) composed of independently developed and managed elements—sensors, command-and-control (C2) systems, effectors, and networks—that must operate coherently to deliver a time-critical mission under uncertainty. In the IAMD operational environment, the decision loop is highly sensitive: sensor observations are fused into tracks, tracks are correlated and identified, engagements are planned and deconflicted across multiple defence systems, and intercept decisions are executed within seconds to minutes. Errors caused by latency, mis-association, configuration mismatch, or misunderstood assumptions can cascade into missed threats or wasted interceptors, with severe consequences for human life and critical assets. At the same time, engineering and sustainment organizations must manage continuous hardware and software evolution, rapid capability fielding, and logistics constraints to maintain readiness over extended operational timelines.

Model-Based Systems Engineering (MBSE) and Digital Engineering (DE) are transforming defence acquisition and sustainment by shifting from document-centric processes to model- and data-driven practices. This transformation creates an opportunity to operationalize digital twins (DTs) not merely as isolated simulations, but as governed decision-support ecosystems connected to operational systems through a digital thread. The Digital Twin Consortium defines a digital twin as an integrated,

data-driven virtual representation of real-world entities and processes with synchronized interaction at a specified frequency and fidelity [1] and identifies DTs as a strategic technology trend alongside the Internet of Things and cloud–edge computing [2]. In aerospace and defence, DTs are typically characterized by two core properties—representation and synchronization—and are expected to deliver value across the system lifecycle, from design and test to operations and sustainment [3].

For complex systems, and especially for IAMD, a DT cannot be a single monolithic model. The "system" of interest is a heterogeneous SoS, often coalition-enabled, operating across security boundaries, and subject to rapidly evolving threats. A single DT attempting to model the entire IAMD SoS at maximum fidelity for both Live-Virtual-Constructive (LVC) integration and operational decision support is neither feasible nor necessary. Instead, an effective IAMD DT must be a federated family of models operating at multiple levels of abstraction and fidelity, dynamically connected to operational and engineering data. High-fidelity physics-based or subsystem models support design analysis and failure investigation, while lower-fidelity abstractions enable rapid trade-off analysis, monitoring, and near-real-time decision support. Crucially, these models must be synchronized to a shared, configuration-controlled system state.

Prior work in air and missile defence addresses many important elements of this problem. Systems engineering perspectives emphasize the need to address increasingly complex threats and operational environments holistically [4]. System-of-systems analyses highlight the importance of networked sensors, shared track pictures, end-to-end performance assessment, and interoperability among distributed components [5]. Modelling and simulation tools such as the Extended Air defence Simulation (EADSIM) are widely used for analysis of alternatives, engagement studies, and multi-tier defence evaluation [6]. Other studies stress that simulation supports training and analysis only when assumptions, limitations, and uncertainty are well understood and communicated [7]. Collectively, this literature provides depth in systems engineering practice, SoS performance assessment, simulation environments, and engagement analysis.

However, much of the existing work focuses on specific slices of the air-defence problem—individual tools, subsystems, analytical studies, or organizational functions. What remains less developed is a holistic DT architecture that links synchronized operational state, MBSE artifacts, controlled digital threads, LVC execution, and traceable decision products across the lifecycle. In practice, engineering models, operational dashboards, simulations, training scenarios, and sustainment data often exist, but they are separated by incompatible tools, schemas, governance processes, and security domains. As a result, engineering baselines may not align with training scenarios, operational analytics may not reflect current configurations, and analytical results based on outdated or undocumented assumptions may not be trusted for high-stakes decisions.

These challenges highlight a key premise of this paper: a DT for IAMD is fundamentally an integration and governance problem, not only a modelling problem. Trustworthy DTs require configuration control, data provenance, interoperability, and stakeholder alignment in addition to executable models. defence DE policy and standards emphasize this point by positioning models and data as authoritative lifecycle artifacts. For example, DoDI 5000.97 frames DE as a shift toward digital models and digital threads to improve traceability, reuse, and consistency across stakeholders [8], while the DoD Digital Engineering Strategy emphasizes integrated digital environments and authoritative digital representations throughout the lifecycle [9]. Prior DT research similarly stresses that value emerges from governed ecosystems of interoperable tools and processes, not from a single "all-in-one" platform [10]–[12].

Within this ecosystem, MBSE artifacts and architectural frameworks play a critical role. SysML provides a standardized language for representing system structure, behaviour, requirements, and relationships, enabling consistency and model exchange across tools [13]. The Department of Defense Architecture Framework (DoDAF) complements SysML by providing viewpoints tailored to operational, technical, and managerial stakeholders, which is particularly important for complex defence SoS [14]. Together, SysML and DoDAF support traceability, communication, and alignment, while executable models instantiate selected portions of the architecture for simulation, trade-off analysis, and decision support. In this paper, SysML and DoDAF define the architectural and traceability logic, while executable Python-based models demonstrate how that logic can be operationalized.

IAMD modernization efforts further illustrate the need for SoS-level DT thinking. Government Accountability Office assessments note that air and missile defence programs spanning multiple organizations and contractors face persistent challenges in requirements stability, technology maturity, testing, and integration [15]. Architectures such as networked battle management and integrated C2 systems underscore that performance depends on interoperability across sensors, networks, C2, and effectors rather than on isolated subsystem metrics [4], [5], [16]. Because interoperability is a dominant risk driver at the SoS level, DT-enabled regression testing, analysis, and configuration-aware assessment can provide significant value.

Live-Virtual-Constructive integration is a critical enabler that connects engineering, testing, training, and operations within a DT ecosystem. LVC environments combine live systems and personnel, virtual simulators, and fully constructive models to scale experimentation while preserving operational relevance. Effective LVC requires interoperability, time synchronization, configuration management, and evidence capture. Prior work emphasizes the importance of characterizing consistency and validity in LVC environments, particularly when they are used for more than training, such as test and analysis [17]. Standards such as the High-Level Architecture (HLA) support simulation interoperability through publish–subscribe mechanisms and common services [18]. For an IAMD DT, LVC provides the practical bridge between engineering fidelity and operational credibility: when the DT uses the same models, data, and scenarios as training and test environments, learning and trust can accumulate across domains rather than fragment.

From this context, several top-level requirements for an IAMD DT ecosystem can be derived. An effective DT must (i) support multiple levels of fidelity and time horizons, from component-level design analysis to real-time SoS monitoring [3], [8], [9]; (ii) manage operational data streams with latency suitable for short-horizon prediction while also incorporating lower-rate sustainment and configuration data [1], [15]; (iii) maintain configuration-controlled baselines and provenance for both data and models, ensuring traceability and repeatability [8], [9]; (iv) define common semantics and data exchange mechanisms to enable interoperability across sensors, C2 systems, simulations, and engineering repositories [12], [14]; (v) implement robust verification, validation, and accreditation (VV&A) processes that communicate uncertainty and avoid opaque “black-box” outputs for high-stakes decisions [3], [17]; (vi) operate across security boundaries while preserving traceability; and (vii) deliver actionable, explainable decision-support products aligned with operator workflows.

Based on this problem formulation, this study addresses three research questions:

- What reference architecture enables a DT of an IAMD SoS to remain synchronized, traceable, and useful across multiple decision contexts and time scales?
- How should LVC environments be integrated into the DT so that data, models, scenarios, and decision products remain coherent across engineering, testing, training, operations, and sustainment?
- How can a synchronized IAMD SoS state be used to initialize executable model logic, evaluate trade-offs under uncertainty, and produce traceable decision-support outputs?

This paper makes three contributions. First, it proposes a reference architecture for an IAMD Digital Twin that integrates data, model, and decision layers through an LVC-compatible digital thread. Second, it defines a traceable decision-support workflow linking synchronized state snapshots to executable model assumptions, uncertainty treatment, and decision products. Third, it demonstrates the architecture through a Python-based operational research demonstrator using an unclassified representative engagement scenario. The demonstrator evaluates four illustrative trade studies—engagement-policy selection, acquisition alternatives, fix-or-fight decision-making, and design sensitivity—not to draw universal conclusions, but to show how synchronized state, model logic, Monte Carlo uncertainty, and traceable decision outputs can be coherently integrated.

The remainder of the paper is organized as follows. Section 2 presents the proposed reference architecture for an IAMD DT. Section 3 introduces the LVC-integrated DT implementation concept. Section 4 describes the Python-based demonstrator. Section 5 presents results from four representative trade-off studies. Section 6 discusses benefits, risks, and open challenges, and Section 7 concludes with implications for future work.

2. Reference Architecture for an IAMD DT

This section introduces a reference architecture for an IAMD SoS DT. The architecture is organized into three tightly coupled layers—a Data Layer, Model Layer, and Decision Layer—that together enable synchronized state awareness, executable analysis, and traceable decision support across multiple time horizons. In addition, the section introduces a traceability framework that links operational needs to system functions, model assumptions, and decision outputs.

The proposed architecture is tool- and vendor-agnostic. Rather than prescribing a specific commercial solution, it defines the functional decomposition, interfaces, and integration logic required to implement an LVC-integrated DT that can evolve incrementally and scale across stakeholders, security domains, and levels of fidelity.

2.1 Architectural Overview

Figure 1 illustrates the proposed IAMD DT ecosystem as an interaction among three elements: the physical IAMD SoS, the DT environment, and the stakeholder community.

The physical SoS comprises the operational elements that generate authoritative state data, including sensors, C2 systems, communication networks, launchers, interceptors, and sustainment systems. These systems operate in real environments and under real constraints, producing telemetry, track data, health status, and configuration changes.

The DT environment ingests these data via the digital thread and organizes them into data services, model services, analytics services, and decision-support products. Critically, the DT is not a single simulation or analytics tool. It is a governed ecosystem that connects operational data, engineering artifacts, executable models, and human decision-makers in a coherent and traceable manner. Stakeholders—operators, maintainers, testers, program managers, and commanders—interact with the DT through role-appropriate views and decision products derived from the same synchronized system state.

2.2 Guiding Principles

To achieve scalability, credibility, and lifecycle relevance, the reference architecture follows five guiding principles:

- MBSE-Centric Context: The DT's core context is the MBSE architecture and requirements baseline—not a standalone simulation [13],
- Layered Modularity: Each layer evolves independently but is integrated through well-defined contracts to enable scalability and modularity,
- Federated Structure: The DT is a federation of subsystem twins (sensor, effector, network, C2) coordinated at the SoS level. This reflects SoS realities and supports incremental adoption,
- Embedded LVC Integration: LVC consumes and produces DT data products, functioning as an execution context for DT models and scenarios rather than as an external lab,
- Transparent Predictions: Every prediction is linked to assumptions, model version, data provenance, and confidence levels [17].

2.3 Layered Architecture

From an architectural perspective, the IAMD DT comprises three primary layers

The Data Layer manages authoritative representations of the system's evolving state. It includes live and near-real-time inputs such as telemetry, track data, environmental context, maintenance events, and configuration changes.

A central construct in this layer is the state snapshot: a time-stamped, configuration-controlled representation of the relevant SoS state. State snapshots provide a reproducible initialization point for simulations, trade studies, and decision analysis, enabling analysts and operators to answer not only "what is happening now?" but also "what would happen if… starting from this exact state?"

The Model Layer organizes both descriptive and executable models and defines how they may be used. It includes:

- MBSE artifacts, such as requirements, interfaces, behaviours, and parameters captured in

SysML;

- SoS architectural views, including DoDAF operational, system, and capability viewpoints that support stakeholder communication and alignment [14];
- Executable models at multiple fidelities, such as engagement-level kinematics, sensor-performance abstractions, communications-delay models, and constructive campaign simulations.

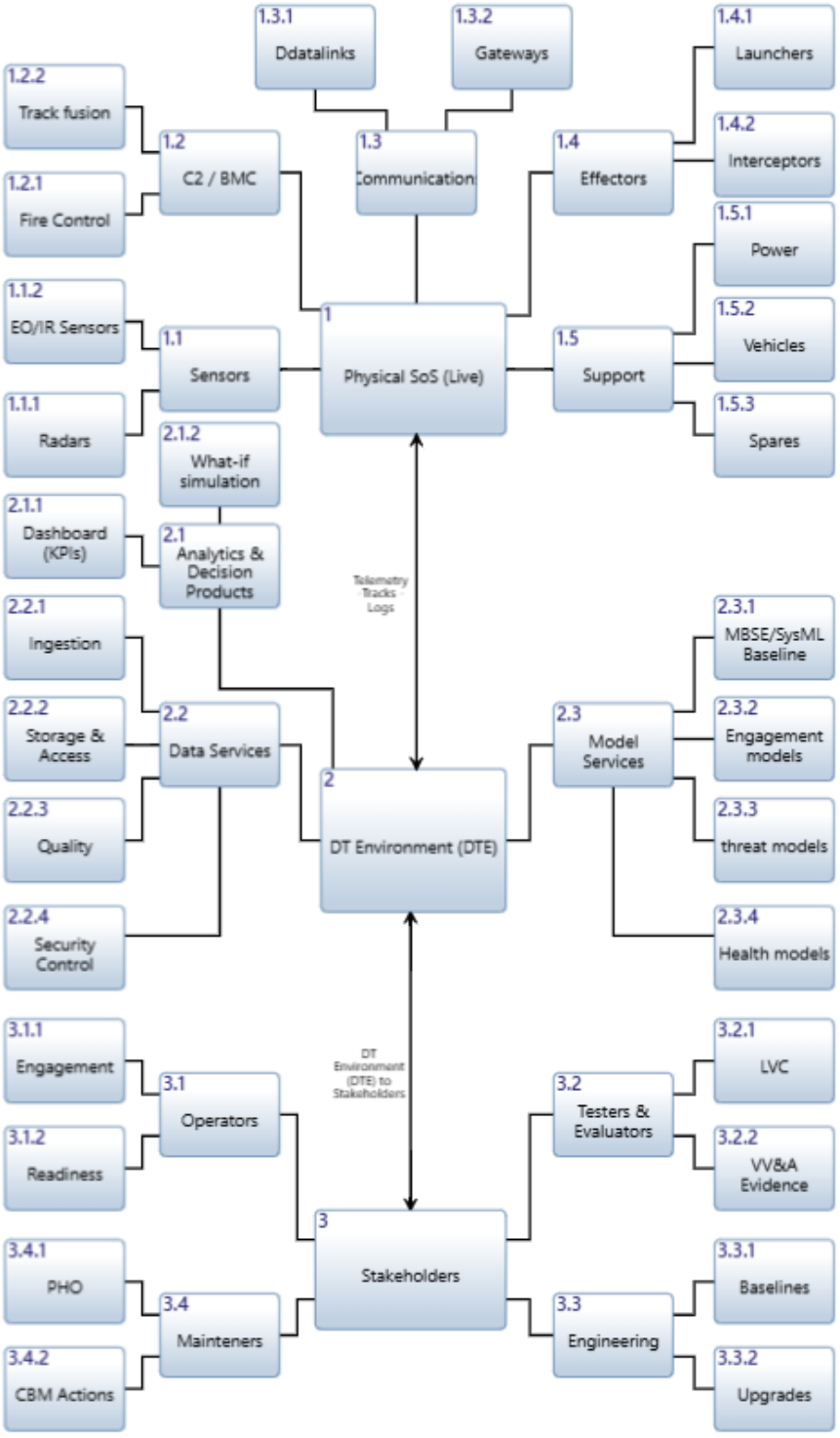


Figure 1 - IAMD DT ecosystem.

The Decision Layer transforms synchronized state and model outputs into actionable decision-support products. These products are organized across three decision horizons:

- Short-horizon (seconds to minutes): anomaly detection, system health alerts, track-quality

degradation indicators;
- Mid-horizon (hours to weeks): readiness forecasts, maintenance prioritization, spares allocation, and risk-informed scheduling;
- Long-horizon (months to years): force-design trade studies, tactic effectiveness across threat ensembles, and capability gap exploration.

2.4 Closed-Loop Decision Support

Figure 2 illustrates how the DT operationalizes decision support as a closed-loop process.

The loop begins with sensing the physical system and synchronizing the DT state. Executable virtual or constructive models are then initialized from that state and executed across alternative courses of action. Results are synthesized into decision recommendations, which in turn influence operational actions, configuration changes, or planning decisions. As new operational evidence is observed, model assumptions, thresholds, and scenario libraries are updated.

This learn-and-act feedback loop is essential: without it, the DT risks becoming stale, miscalibrated, or disconnected from reality. The logic shown in Figure 2 underpins the executable experiments presented later in the Python-based demonstrator.

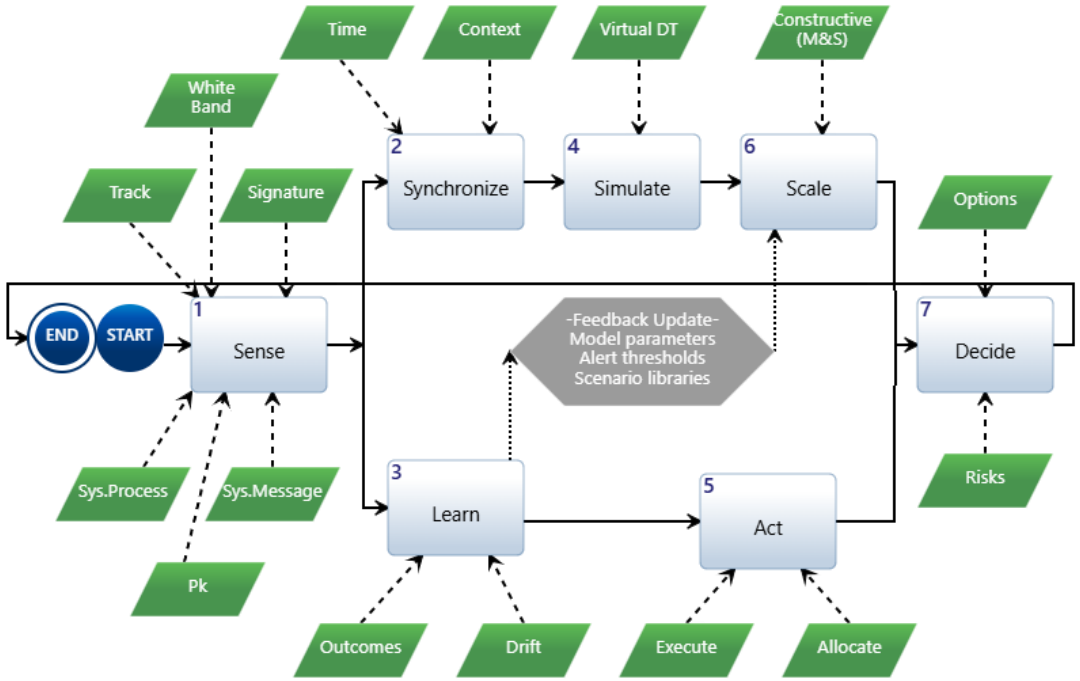


Figure 2 - Decision support loop.

2.5 Traceability Framework

Given the scale, heterogeneity, and data volume of IAMD systems, the DT requires a robust and explicit traceability framework. At a minimum, this framework includes:

- Unique identifiers for assets, configurations, interfaces, and model elements;
- A controlled vocabulary for measures and performance metrics (e.g., track quality, time-to-intercept, probability of kill, availability, mean time to repair);
- Explicit links between operational needs, system functions, architectural elements, model assumptions, verification evidence, and operational metrics.

The objective is to ensure that every decision product can be traced to:

- the specific state snapshot used,
- the model versions and assumptions applied,
- and the stakeholder need or requirement it supports [8], [9].

Figure 3 illustrates this concept as a traceability spine that prevents DT outputs from becoming disconnected or non-actionable analysis artifacts.

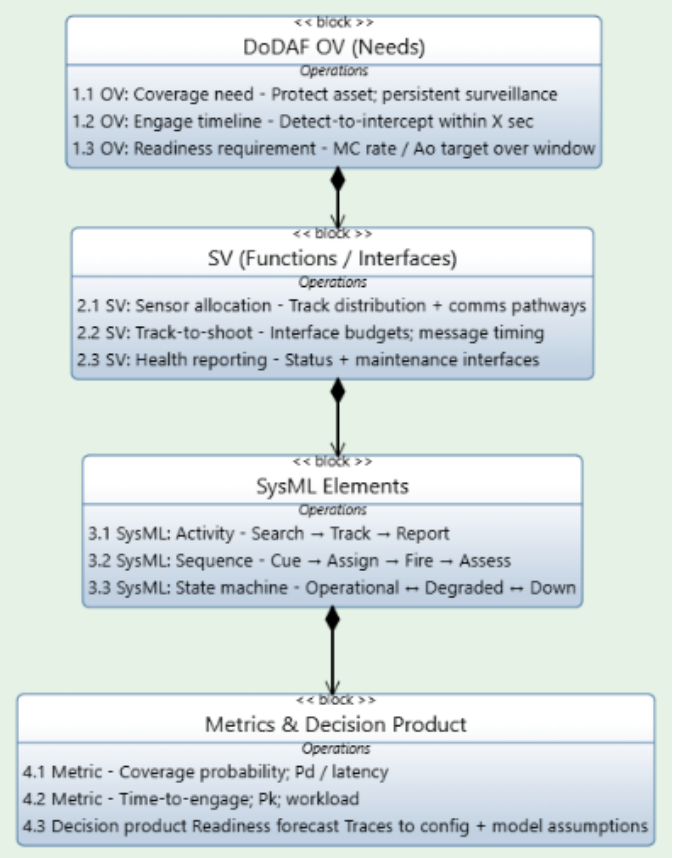


Figure 3 - Traceability spine.

Operational needs are mapped through system functions and interfaces to model elements, assumptions, metrics, and evidence. As a result, a policy recommendation, readiness forecast, or design trade result can always be traced back to its data sources, modelling logic, and decision context—preserving credibility and enabling informed human judgment.

3. LVC-integrated DT Implementation Concept

This section describes how the proposed IAMD Digital Twin (DT) architecture is operationalized through LVC integration. In this concept, LVC is not treated as a standalone training capability or isolated simulation laboratory. Instead, it functions as a federated execution context for DT models, scenarios, and evidence generation, enabling coherence across operations, training, testing, and engineering analysis.

The core objective of LVC integration is to ensure that operational data, simulation models, training scenarios, and analytical evidence remain synchronized, traceable, and reusable across lifecycle activities. Achieving this requires well-defined interfaces between live systems, the DT, and virtual and constructive execution environments, as well as disciplined scenario and time-management strategies.

The LVC-integrated DT is implemented through three primary interfaces that connect operational reality to executable models and scalable analysis

- Live-to-Twin Interface - Streams operational telemetry, track data, system health, and configuration updates from the physical IAMD SoS into the DT. This interface produces configuration-controlled state snapshots that serve as authoritative initial conditions for all subsequent virtual and constructive execution.
- Twin-to-Virtual Interface - Initializes virtual simulations—such as operator-in-the-loop trainers or mission rehearsal environments—with synchronized DT state, selected model fidelity, and scenario parameters. Results are exported back to the DT as structured, versioned data products rather than informal outputs.
- Twin-to-Constructive Interface - Generates ensembles of scenarios, parameter variations,

and Monte Carlo inputs derived from the same baselines and assumptions used in engineering and training. This interface enables large-scale trade studies and campaign-level analysis without manual reconfiguration.

Together, these interfaces reduce duplication and inconsistency across tools and organizations. When training scenarios, test cases, and operational analytics are derived from a shared DT baseline, the DT becomes a common analytical language for the IAMD SoS rather than a collection of disconnected simulations.

LVC integration is implemented as a federated execution environment in which live systems, virtual simulators, and constructive models exchange data through standardized interfaces. Publish–subscribe mechanisms, such as those provided by the High-Level Architecture (HLA), offer a practical means for interconnecting heterogeneous simulators and live data sources [18].

Existing IAMD modelling and simulation environments, such as EADSIM, demonstrate that simulation can support integrated analysis across sensors, effectors, and engagement conditions [6]. However, prior work also emphasizes that simulation results are credible only when assumptions, intended use, and limitations are explicitly documented and communicated [7]. For this reason, LVC data exchange within the DT is grounded in mission-relevant entities, including tracks, sensor reports, engagement assignments, effector states, network delays, and timing constraints, all referenced through controlled schemas and definitions (Figure 4).

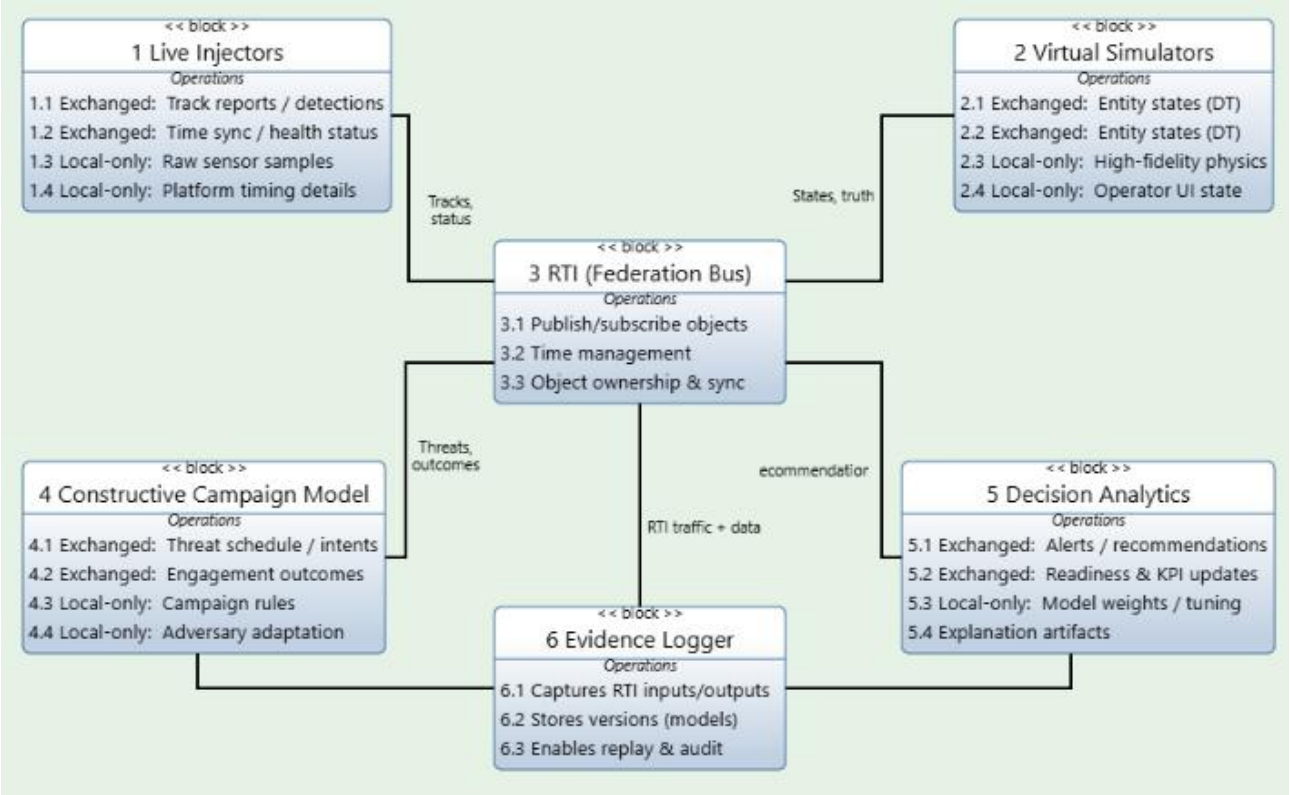


Figure 4 - LVC federation data exchange.

A critical architectural decision concerns time synchronization across execution contexts. The DT supports multiple time-management modes, each aligned with different decision objectives:

- Real-time execution, supporting operator training, mission rehearsal, and live decision aids;
- Time-scaled execution, enabling accelerated or slowed testing, regression analysis, and tactics development while preserving causal relationships;
- Faster-than-real-time execution, supporting constructive campaign-level analysis, force-design studies, and uncertainty exploration through large ensembles.

By explicitly managing time as a configurable property of execution, the DT allows the same models and scenarios to support multiple decision horizons without reimplementation.

In this architecture, scenarios are treated as configuration-controlled digital artifacts rather than ad

hoc training inputs. The DT scenario library stores and manages:

- Threat models and environmental parameters;
- Blue-force configurations and tactics templates;
- Training objectives, evaluation criteria, and performance metrics;
- Scenarios derived from operational data and after-action evidence.

Live systems contribute synchronized state and configuration data, virtual simulations evaluate behaviour at selected levels of fidelity, and constructive models scale analysis across large scenario ensembles or Monte Carlo runs. An evidence logger records model versions, parameter settings, inputs, outputs, and assumptions for every LVC execution. This evidence capture ensures that LVC runs produce repeatable analytical artifacts rather than disposable training events.

The LVC-integrated DT supports two reinforcing feedback loops. The training loop updates scenarios, tactics, and evaluation metrics based on operational experience and observed performance. The engineering loop uses regression suites and controlled scenario sets to assess the impact of software changes, configuration updates, or new capabilities on end-to-end SoS behaviour. Together, these loops help ensure that modernization decisions improve performance without introducing unintended interactions or degrading SoS effectiveness [12].

By embedding LVC within the DT rather than treating it as a separate enclave, the proposed implementation preserves consistency, improves trust, and enables continuous learning across engineering, training, operations, and sustainment.

4. DT-Enabled Operational Research Platform

This section instantiates the proposed DT architecture using a Python-based operational research platform. The purpose is not to present a validated or authoritative tactical model, but to demonstrate how a synchronized IAMD SoS state can be frozen into a DT snapshot, used to initialize executable logic, explored through Monte Carlo analysis, and transformed into traceable decision-support products.

The platform illustrates how physical-system configuration, modelling assumptions, stochastic uncertainty, and decision logic are explicitly connected. This section describes the model structure, scenario definition, assumptions, and evaluation metrics, that are then used in four representative trade-off studies—engagement policy, acquisition alternatives, fix-or-fight decisions, and design sensitivity—that exercise the DT workflow.

The demonstrator corresponds to the Sense–Synchronize–Simulate–Scale–Decide branch of the decision-support loop shown in Figure 2. It demonstrates how a synchronized IAMD SoS configuration is ingested into the DT, used to initialize virtual and constructive execution, and evaluated under uncertainty to produce decision outputs annotated with assumptions, parameter values, and confidence intervals.

The model architecture comprises five functional elements:

- Scenario generation generates attacker launch points, intended impact points, defender radar locations, launcher locations, and critical-zone definitions,
- State synchronization freezes all scenario parameters and geometry into a DT state snapshot, ensuring that all alternatives are evaluated against the same initial conditions,
- Detection and tracking logic determines radar visibility, trajectory coverage, track-quality modifiers, and the contribution of multiple radars to track quality,
- Engagement-policy logic assigns interceptor shots subject to launcher range, interceptor inventory, and engagement rules,
- Monte Carlo aggregation samples equipment malfunctions and stochastic kill outcomes and reports mean values and 95% confidence intervals for each performance metric.

The IAMD model is structured around the interaction between an attacker attempting to penetrate defences and a defender protecting designated assets (Figure 5). Both attacker and defender regions are represented as 100 km × 100 km squares. Within the defender region, a 30 km × 30 km critical zone represents strategic assets and high-value population centres.

In the demonstrator interface, raids of 1 to 100 threats can be generated. For the reference experiment presented in this paper, the synchronized raid contains 30 threats. Attacker launch points

are randomly distributed in the attacker region, and intended impact points are split evenly: half are randomly assigned within the critical zone and half outside it. Leakage into the critical zone is given a weight of five, while leakage outside the zone receives a weight of one, reflecting differential consequence severity.

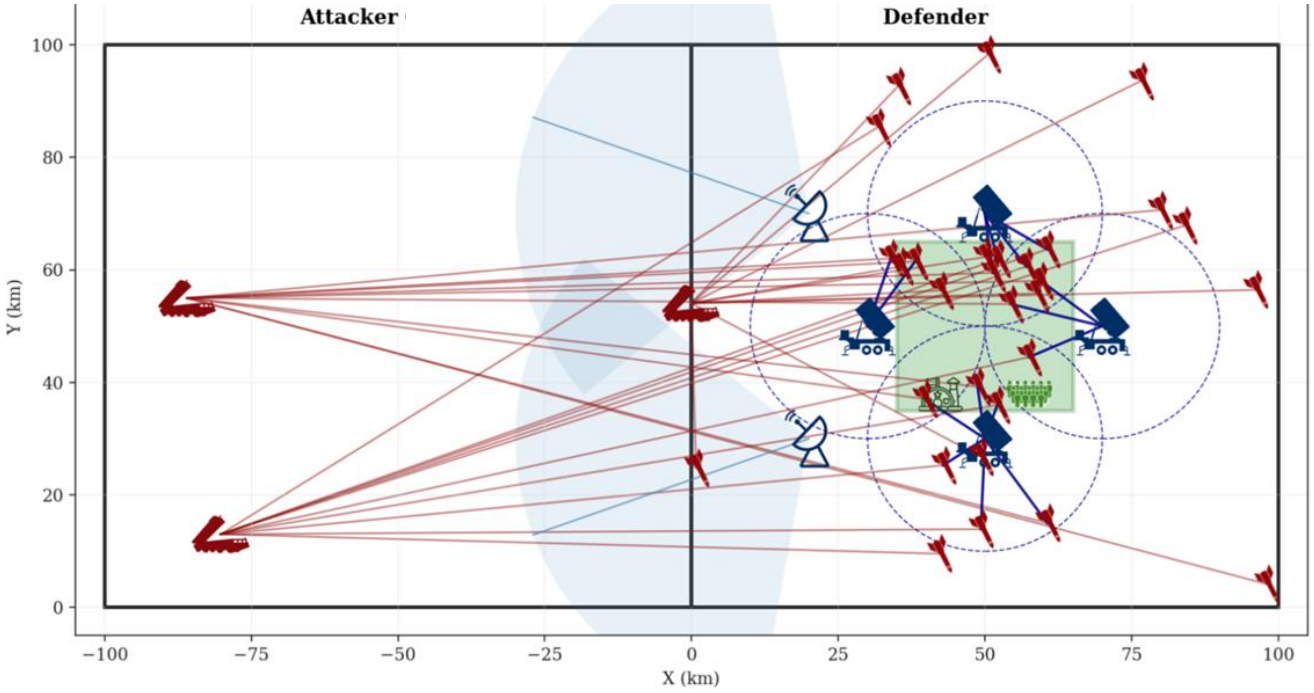


Figure 5 - Map of the physical-system configuration.

The defender consists of radars, launchers, and interceptors. Radars provide geometric surveillance coverage and detect threats when a sufficient portion of the threat trajectory falls within their field of view. Launchers represent the effector layer, each with a finite interceptor inventory and a fixed engagement range; only threats whose intended impact points fall within this range can be engaged.

Once synchronized, the DT uses the same physical-system configuration to evaluate detection, assignment, and engagement outcomes under different policies, readiness levels, acquisition alternatives, and design parameters. Table 1 summarizes the synchronized state snapshot used to initialize the DT experiments.

As shown in Figure 5, on the attacker side, launcher icons show threat origin points, with red trajectory lines extending to intended impact points within the defender region. On the defender side, radar icons depict sensor coverage (range and field of view in blue), launcher icons indicate effector positions, and dashed circles represent engagement ranges. Blue engagement lines illustrate interceptor assignments generated under a selected engagement policy. Together, these elements allow the reader to visualize the synchronized DT state, including threat geometry, sensor coverage, launcher reachability, critical-zone exposure, and engagement opportunities.

The model demonstrates a DT-enabled workflow that explicitly captures radar detection, track-quality effects, launcher constraints, interceptor allocation, equipment malfunctions, and stochastic engagement outcomes, while keeping all results traceable to stated assumptions.

The baseline single-shot probability of kill is modelled as a function of launcher-to-target distance

$$P_k^{base}(d) = \max(0, \min(1, 1 - d/40)) \tag{1}$$

where d is the distance (km) from the launcher to the target aiming point.

Track quality depends on the maximum trajectory length observed within radar field of view

$$\mathrm{TQF}(L_{max}) = \begin{cases} 0.70, & L_{max} < 20 \text{ km} \\ 0.85, & 20 \le L_{max} < 35 \text{ km} \\ 1.00, & L_{max} \ge 35 \text{ km} \end{cases} \tag{2}$$

A modest improvement is applied when multiple radars, n_r, contribute to tracking

$$\mathrm{MRF}(n_r) = \begin{cases} 1.00, & n_r = 1 \\ 1.05, & n_r = 2 \\ 1.10, & n_r \geq 3 \end{cases} \quad (3)$$

Combining these factors yields the final single-shot probability of kill

$$P_k^{final} = min\left(1, P_k^{base} \cdot \mathrm{TQF} \cdot \mathrm{MRF}\right) \quad (4)$$

For n independent interceptor shots against the same threat, the cumulative kill probability is

$$P_k(n) = 1 - (1 - P_k^{final})^n \quad (5)$$

Table 1 - Synchronized DT state snapshot.

Layer	Configuration item	Synchronized DT value	Model range possible value	Decision relevance
Scenario load	Number of targets in synchronized raid	30	1-100	Threat set fixed after sync
	Number of targets inside critical zone	15	0-30	Weighted leakage penalty applies
	Number of targets outside critical zone	15	0-30	Normal leakage weight
Sensor layer	Number of radars	2	1-5	Current physical-system radar count
	Radar capacity	50	Fixed baseline	Target track and interceptor data-link object capacity per radar
	Total radar capacity	100	50-250	Radars × capacity per radar
	Radar range (km)	50	Fixed baseline	Detection geometry parameter
	Radar field of view (deg)	120	Baseline 120; design study varies it	Search-sector design parameter
	Detected trajectories before failures	23	0-30	Geometry-only visibility at sync
	Undetected trajectories before failures	7	0-30	Geometry-only blind spots at sync
Effector layer	Number of interceptor launchers	4	0-10	Current physical-system launcher count
	Number of interceptors per launcher	20	1-30	Magazine depth per launcher
	Total number of interceptors	80	1-300	Launchers × interceptors per launcher
	Engagement range (km)	20.0	Baseline 20; design study varies it	No engagement beyond this range
Geometry	Number of the launchers within 0-10 km	6	0-30	Threat aiming points close to a launcher
	Number of the launchers within 10-20 km	14	0-30	Threat aiming points within nominal engagement range
	Number of the launchers within farther than 20 km	10	0-30	Threat aiming points outside nominal engagement range
Readiness	Radar malfunction rate	0.05	0.00-0.20	Sampled inside Monte Carlo runs
	Launcher malfunction rate	0.05	0.00-0.20	Sampled inside Monte Carlo runs
	Interceptor malfunction rate	0.05	0.00-0.20	Sampled inside Monte Carlo runs
DT evaluation	Number of Monte Carlo runs	2000	100-5000	Same synchronized snapshot, randomized outcomes
	Critical-zone leakage penalty	x5	Fixed baseline	Higher penalty for defended assets / population zone

A threat is considered detectable only if at least 10 km of its trajectory lies within at least one radar field of view. Radar and launcher malfunctions are sampled per platform in each Monte Carlo run, while interceptor malfunctions are sampled per shot. Thus, all runs share identical initial geometry but differ in realized equipment states and probabilistic engagement outcomes.

Mission risk is expressed in terms of air-defence leakage, defined as the weighted number of threats that pass through the defence without interception

$$L = \sum_i w_i \, I(leak_i) \quad (6)$$

where $I(leak_i) = 1$ if threat i leaks through the defence and 0 otherwise, $w_i = 5$ for leakage into the critical zone, and $w_i = 1$ elsewhere. The higher value indicates that the defensive "shield" has been partially breached, allowing threats to reach their intended targets.

All reported metrics are stochastic and therefore presented with 95% confidence intervals

$$CI_{95} = \bar{x} \pm 1.96 \frac{s}{\sqrt{N}} \quad (7)$$

where $\bar{x}$ is the sample mean, s is the sample standard deviation, and N is the number of Monte Carlo runs.

To demonstrate DT-enabled trade-off analysis, the IAMD SoS is evaluated under three engagement policies:

- Full Coverage: one interceptor allocated to every detected and reachable threat;
- Shoot–Shoot: two interceptors for threats aimed at the critical zone and one interceptor elsewhere;
- Inventory-Preserving: one interceptor for critical-zone threats; outside the critical zone, engagement occurs only when $P_k^{\text{final}} > 0.8$.

These policies illustrate how a synchronized DT state, executable policy logic, and uncertainty treatment interact to produce comparative, traceable decision-support products. The resulting trade studies show how policy choices affect leakage, interceptor expenditure, robustness to malfunctions, and uncertainty—providing an explicit link from configuration and assumptions to decision-relevant outcomes.

5. Trade-Off Experiments

This section presents four representative trade-off experiments generated from the synchronized state snapshot and executable model logic described in Section 4. Together, they demonstrate how a single DT architecture can support operational, acquisition, readiness, and design decisions while preserving traceability to a common set of inputs, assumptions, and uncertainty treatments. All results are derived from the same synchronized threat geometry and physical-system configuration unless explicitly varied.

5.1 Engagement Policy Selection

The first experiment compares three engagement policies—Full Coverage, Shoot–Shoot, and Inventory-Preserving—against the same synchronized threat raid. Monte Carlo results for this comparison are presented in Table 2. As shown, the *Shoot–Shoot* policy achieves the lowest weighted leakage (42.9 ± 0.5) and the lowest critical-zone leakage (6.0 ± 0.1), but it also consumes the most interceptors (24.5 ± 0.2). This outcome reflects the policy's prioritization of critical-zone threats with multiple shots. *Full Coverage* uses fewer interceptors than *Shoot–Shoot* but produces higher weighted leakage. *Inventory-Preserving* achieves the highest interceptor efficiency (0.63 ± 0.01) by firing selectively, but at the cost of the largest weighted leakage.

Table 2 - Monte Carlo results for engagement policy selection.

Policy	Recommended	Weighted leakage (95% CI)	Critical leakage (95% CI)	Neutralized threats (95% CI)	Interceptors used (95% CI)	Efficiency ratio (95% CI)	Detected trajectories
Full Coverage		54.5 ± 0.6	8.3 ± 0.1	8.9 ± 0.2	14.0 ± 0.1	0.630 ± 0.008	22.073
Shoot-Shoot	Yes	42.9 ± 0.5	6.0 ± 0.1	11.2 ± 0.1	24.5 ± 0.2	0.455 ± 0.004	22.218
Inventory Preserving		56.9 ± 0.6	8.4 ± 0.1	6.6 ± 0.1	10.4 ± 0.1	0.63 ± 0.01	22.001

Figure 6 depicts these results as a mission–resource frontier: reductions in mission risk require increased interceptor expenditure. Figure 7 decomposes leakage into total, critical-zone, and weighted components, showing that *Shoot–Shoot* derives its advantage primarily from reducing critical-zone leakage.

This experiment illustrates the role of the DT decision layer: it quantifies the trade-off between mission risk and resource consumption and allows policy preferences to be interpreted explicitly in terms of acceptable leakage versus interceptor expenditure.

5.2 Support of Acquisition Decisions

The second experiment demonstrates how the DT can support acquisition reasoning by comparing investments in additional radars versus additional launchers. Starting from the same synchronized system state, the baseline configuration is compared with alternatives that add one or two radars or one or two launchers. A normalized acquisition cost of 5.0 is assigned to each added radar and 1.0 to each added launcher; these unit costs are configurable model parameters. All acquisition cost

values are expressed in normalized cost units rather than monetary currency. In this demonstrator, one added launcher is used as the reference cost unit of 1.0, and one added radar is assumed to cost 5.0 normalized units. These values are illustrative model parameters used only for relative comparison among alternatives.

Table 3 summarizes the acquisition alternatives evaluated, including force composition, normalized acquisition cost, weighted leakage, leakage improvement relative to the baseline, and improvement per unit cost.

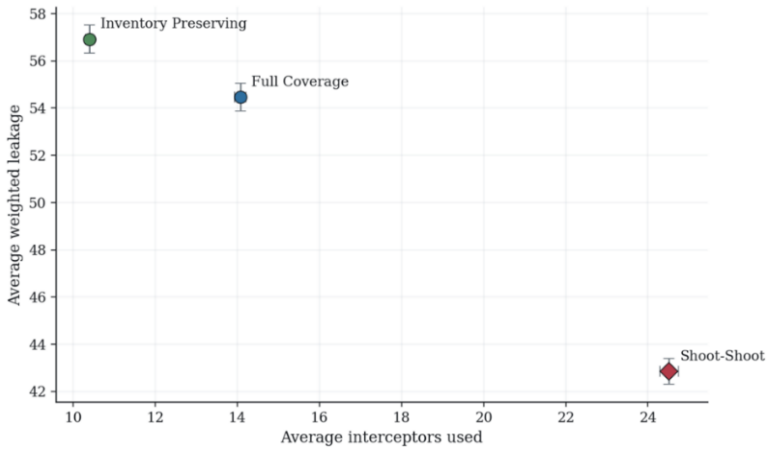


Figure 6 - Mission-resource frontier for engagement policy selection.

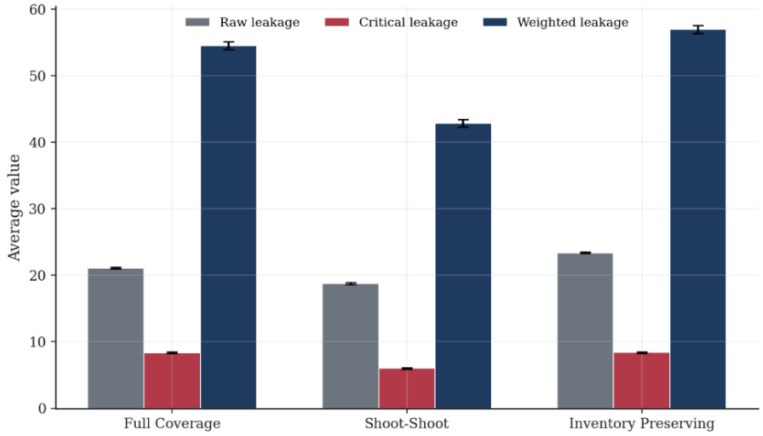


Figure 7 - Leakage outcomes by engagement policy.

Table 3 - Acquisition trade-off results: radars versus launchers.

Alternative	Radars	Launchers	Acquisition cost	Weighted leakage (95% CI)	Weighted leakage improvement	Improvement per cost
Baseline force	2	4	0	43.6 ± 0.6	0	0
Add 1 radar	3	4	5	27.7 ± 0.9	15.9	3.17
Add 2 radars	4	4	10	25.9 ± 0.6	17.7	1.77
Add 1 launcher	2	5	1	42.4 ± 0.6	1.2	1.22
Add 2 launchers	2	6	2	42.1 ± 0.6	1.5	0.74

As shown in Table 3, radar additions produce the largest absolute reduction in weighted leakage. Adding one radar reduces weighted leakage from 43.6 ± 0.6 to 27.7 ± 0.9, while adding two radars further reduces it to 25.9 ± 0.6. In contrast, adding launchers yields only marginal improvements, and confidence intervals show that launcher-only alternatives remain close to the baseline.

Figures 8 and 9 illustrate the trade-off from complementary perspectives. Figure 8 shows weighted leakage versus normalized acquisition cost, while Figure 9 shows leakage reduction per cost unit.

In this representative geometry, radar additions yield both the largest absolute improvement and the highest improvement per unit cost, indicating that system performance is constrained more by detection and tracking capacity than by interceptor quantity. This result should not be interpreted as

a general procurement recommendation. It is a scenario-specific decision product derived from a synchronized DT snapshot and normalized cost assumptions, demonstrating how the DT connects physical configuration changes to measurable acquisition trade-offs.

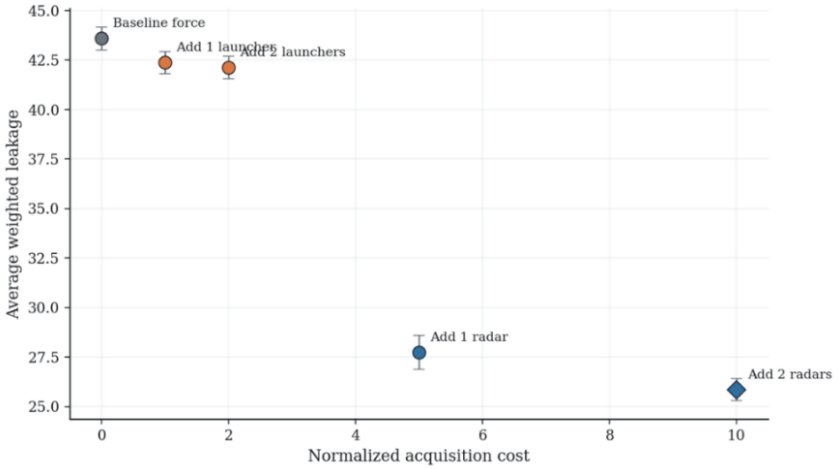


Figure 8 - Normalized acquisition cost versus weighted leakage.

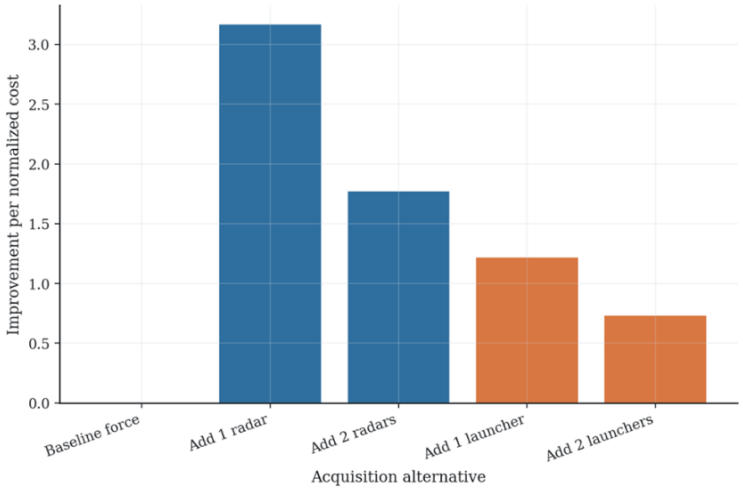


Figure 9 - Marginal value of acquisition alternatives.

5.3 Support of Fix-or-Fight Decision

The third experiment illustrates how the DT supports readiness and maintenance decisions in response to radar degradation. Each radar is assigned a health state $h_r \in [0,1]$. During a Monte Carlo run, each successful detection reduces radar health by $\delta = 0.02$

$$h_{r,t+1} = \max(0,\ h_{r,t} - \delta) \tag{8}$$

If health falls below a selected maintenance threshold τ, a repair action restores full health at a normalized cost of C_{repair} =0.2. The repair cost is also expressed in normalized cost units; a value of 0.2 represents an illustrative relative repair burden, not a monetary estimate.

$$\text{If}\ \ h_{r,t+1} < \tau,\ \ \text{then}\ \ h_{r,t+1} = 1.0\ \ \text{and}\ \ C_{\text{repair}} = C_{\text{repair}} + 0.2 \tag{9}$$

Radar health acts as a multiplier on engagement effectiveness, modifying the final probability of kill (see Equation 4) as

$$P_k^{final} = min\left(1, P_k^{base} \cdot \text{TQF} \cdot \text{MRF} \cdot h_{r,t}^{+}\right) \tag{10}$$

In this modified equation,· $h_{r,t}^{+}$ is the radar health state after degradation and any repair action has been applied.

Three maintenance strategies are evaluated: reactive ($\tau = 0.5$), moderate ($\tau = 0.7$), and aggressive ($\tau = 0.9$), along with a no-repair baseline.

Table 4 summarizes the evaluated strategies, including repair thresholds, weighted leakage, detected trajectories, average final radar health, number of repair actions, normalized repair cost, and cost per leakage avoided.

Table 4 – Fix-or-fight trade-off results: radar degradation and repair.

Strategy	Repair threshold	Weighted leakage (95% CI)	Detected trajectories	Average final radar health	Repair actions	Repair cost	Cost per leakage avoided
No repair	—	45.3 ± 0.6	19.708	0.75	0	0	0
Reactive repair < 0.50	0.500	45.0 ± 0.6	19.757	0.76	0	0	0
Selected repair < 0.70	0.700	45.2 ± 0.6	19.738	0.80	0.3	0.06	0.47
Aggressive repair < 0.90	0.900	44.4 ± 0.6	21.394	1.00	4.5	0.91	0.98

Figures 10 and 11 show the relationship between repair cost, radar health, and mission risk.

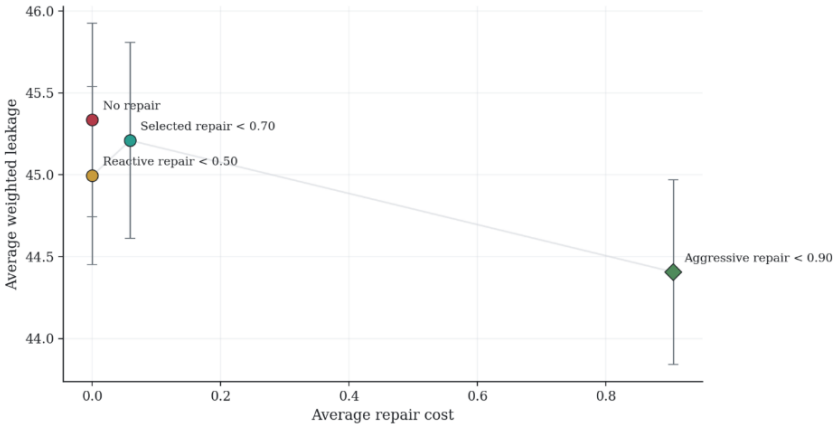


Figure 10 - Maintenance cost versus weighted leakage.

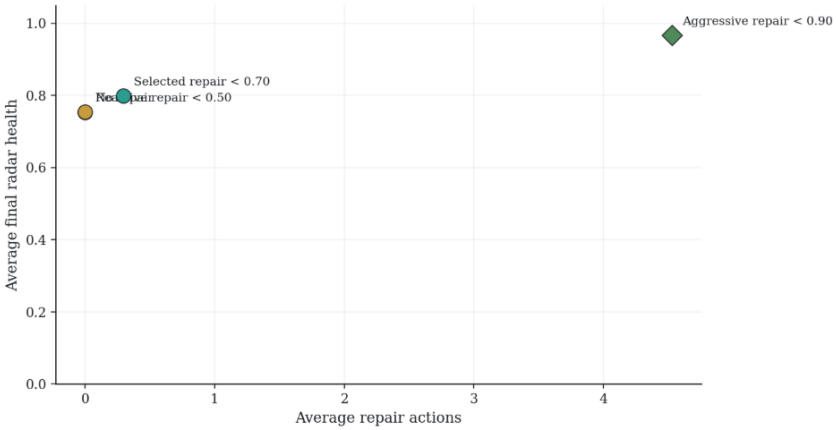


Figure 11 - Repair actions versus final radar health.

The results show only modest separation in weighted leakage among no-repair, reactive, and moderate strategies in this low-load (30-target) scenario. The aggressive strategy yields the lowest weighted leakage and highest radar health but at significantly higher repair cost and operational burden. This experiment demonstrates how the DT links degradation models, repair policies, cost, and mission risk—even when differences are small—allowing readiness decisions to be evaluated transparently under uncertainty.

5.4 Support of Design Decision

The final experiment demonstrates DT support for design sensitivity analysis. Two design levers are varied: increasing radar field of view (by 20° and 40°) and increasing interceptor engagement range (by 5 km and 10 km). A combined alternative modifies both parameters. All alternatives are evaluated against the same synchronized threat geometry.

Table 5 summarizes the evaluated design alternatives, including radar field of view, interceptor

engagement range, number of detected trajectories, resulting weighted leakage, weighted-leakage improvement relative to the baseline, and the corresponding detection gain. Figures 12 and 13 visualize detection opportunity and leakage reduction.

Increasing radar field of view produces substantially larger reductions in weighted leakage than increasing interceptor range alone. The Radar +40° alternative reduces weighted leakage by 17.0 relative to the baseline, whereas extending interceptor range by 10 km reduces weighted leakage by only 2.4. The combined alternative provides intermediate benefit. These results suggest that, for this synchronized snapshot, detection opportunity is the dominant design constraint.

As with previous experiments, this outcome is scenario-dependent and not intended as a general design recommendation. Its value lies in demonstrating that the DT can connect design parameters to mission-risk metrics under controlled assumptions and shared initial conditions.

Table 5 - Design trade-off results: radar field of view versus interceptor range.

Alternative	Radar FOV (deg)	Interceptor range (km)	Detected trajectories	Weighted leakage (95% CI)	Weighted leakage improvement	Detection gain
Baseline design	120	20	22.2	42.8 ± 0.5	0.0	0.00
Radar FOV +20 deg	140	20	23.5	40.1 ± 0.5	2.7	1.26
Radar FOV +40 deg	160	20	29.2	25.8 ± 0.6	17.0	7.00
Interceptor range +5 km	120	25	22.1	40.8 ± 0.6	2.0	-0.07
Interceptor range +10 km	120	30	22.2	40.4 ± 0.5	2.4	0.00
Combined FOV +20 deg / range +5 km	140	25	23.4	38.0 ± 0.4	4.8	1.20

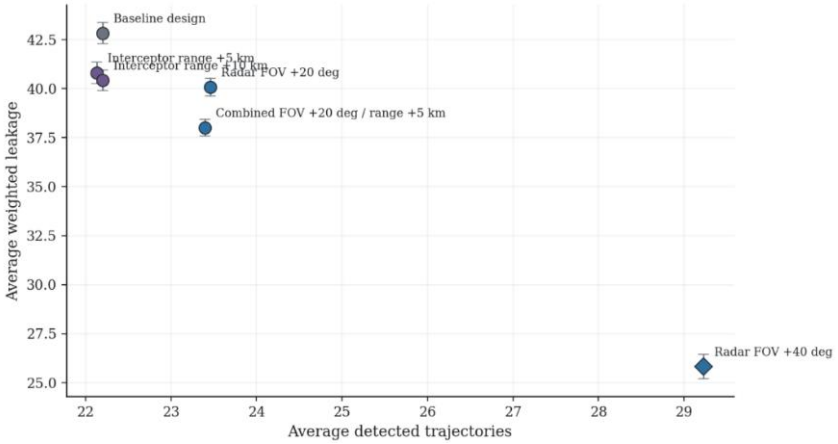


Figure 12 - Detection opportunity and weighted leakage.

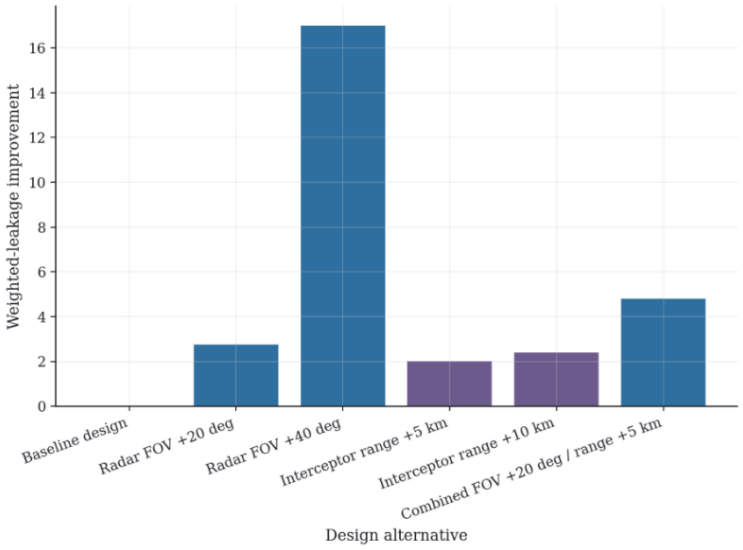


Figure 13 - Weighted-leakage reduction by design alternative.

6. Interpretation, Benefits, Limitations, and Open Challenges

The experiments presented in Section 5 should be interpreted as an architecture demonstrator, not as a finalized tactical or operational IAMD model. Their primary purpose is to address the motivating problem stated in Section 1: demonstrating how a synchronized system state, executable model logic, uncertainty treatment, and decision outputs can be integrated into a reproducible, traceable DT workflow.

The Python-based operational research platform directly instantiates the reference architecture introduced in Sections 2 and 3. The synchronized physical-system configuration corresponds to the data layer; the engagement, readiness, and degradation logic correspond to the model layer; and the trade-off tables and figures correspond to the decision layer. The Monte Carlo experiments represent constructive exploration initialized from a common, configuration-controlled state snapshot. Across engagement policy, acquisition, maintenance, and design studies, the same synchronized baseline is reused, enabling consistent comparison across otherwise disparate decision contexts.

The principal value of the demonstrator lies in its demonstration of an end-to-end DT decision-support workflow. A physical-system configuration is captured and synchronized; alternative decisions are evaluated against that same initial state; assumptions are explicitly declared; and uncertainty is communicated using confidence intervals rather than point estimates. Traceability is an integral part of this value: each result can be linked to the specific state snapshot, model assumptions, parameter values, and decision context that produced it. This directly supports reproducibility, post-decision review, and disciplined use of modelling evidence.

More broadly, the experiments illustrate how an LVC-integrated IAMD DT can reduce duplicated effort across engineering, testing, training, operations, and sustainment. The same models, scenarios, and evidence packages support multiple classes of decisions. This reuse strengthens feedback loops from operations to design and sustainment, enabling anomalies, near-misses, performance drift, and trade-off results to be traced back to configuration data and modelling assumptions. LVC integration also enables more frequent, lower-risk experimentation, allowing tactics, readiness policies, and design alternatives to be explored virtually or constructively before committing to expensive or limited live events.

The principal limitation of the current implementation is model fidelity. To emphasize architectural clarity and traceability, the demonstrator employs simplified geometry, abstract radar sectors, coarse track-quality effects, normalized cost parameters, and independent interceptor-shot outcomes. It does not model high-fidelity kinematics, electronic warfare, communications latency, command-and-control timing, reload constraints, target classification uncertainty, or classified system performance characteristics. Consequently, the numerical results are scenario-specific illustrations, not operational conclusions.

These limitations are intentional. A higher-fidelity implementation could replace individual model components while preserving the same data–model–decision structure, synchronization logic, uncertainty treatment, and VV&A approach. In this sense, the experiments validate the *architecture and workflow*, not the tactical realism of the underlying models.

Despite its benefits, implementing an operational IAMD DT presents several challenges. Data quality and governance are persistent risks: without disciplined configuration management and authoritative data ownership, a DT can devolve into an inconsistent or outdated snapshot, undermining stakeholder trust. A related concern is over-trust, in which decision makers treat DT outputs as definitive even when models operate outside their intended validity range or rely on simplifying assumptions.

LVC integration introduces additional risks. Time-synchronization errors, hidden assumptions embedded in simulation components, or mismatched representations across live, virtual, and constructive domains can lead to misleading results if not clearly understood and documented [17]. Cybersecurity is also a critical concern: compromised DT interfaces could corrupt decision products, expose sensitive information, or weaken confidence in the IAMD digital thread.

Building on the proposed architecture and demonstrator results, several research areas emerge as particularly important:

- Interoperability at SoS scale. Developing lightweight, robust exchange protocols and semantic schemas that operate across vendors, programs, and security domains without creating unmanageable integration complexity. This is especially critical in IAMD, where SoS performance depends on coordination among distributed sensors, shared track pictures, C2 functions, and engagement assets [5].
- Human trust and explainability. Designing decision-support products that include rationale, assumptions, uncertainty bounds, and intended-use limits in forms aligned with operator workflows. Decision support is effective only if humans can understand, challenge, and appropriately trust the output.
- Continuous accreditation. Establishing VV&A processes that evolve with software-intensive systems, changing threat assumptions, and rapidly updated model versions, while maintaining credibility and stakeholder confidence.

7. Conclusion

This paper presented a DT reference architecture for IAMD SoS, with emphasis on LVC integration, MBSE-based traceability, and decision support across operational, acquisition, readiness, and design contexts. The architecture organizes the DT into three coordinated layers: a data layer that captures synchronized system state, a model layer that manages descriptive and executable representations at multiple levels of fidelity, and a decision layer that transforms model outputs into traceable, uncertainty-aware decision products. A Python-based operational research demonstrator instantiated this architecture by mapping synchronized state snapshots to the data layer, engagement and readiness logic to the model layer, and comparative trade-off results to the decision layer. Monte Carlo experiments demonstrated how the same synchronized IAMD scenario can support engagement-policy evaluation, acquisition reasoning, fix-or-fight decisions, and design sensitivity analysis while preserving consistency, provenance, and comparability. The primary contribution of this work is not the specific numerical outcomes of any individual trade-off, which are inherently scenario-dependent and based on simplified assumptions. Rather, the contribution is the reproducible DT workflow and reference architecture that explicitly links synchronized state, executable model logic, uncertainty treatment, and decision products. By making assumptions, model versions, and confidence bounds explicit, the approach supports disciplined use of modelling evidence and reinforces trust in DT-enabled decision support. Future work should focus on replacing simplified model components with higher-fidelity representations, integrating additional live and virtual data sources, strengthening continuous VV&A processes, and expanding evidence-logging mechanisms so that each decision product includes clear intended-use limits. These extensions would move the proposed architecture toward a more credible, continuously updated digital twin environment capable of supporting lifecycle decision-making in operational IAMD SoS.

Contact Author Email Address

The contact author email address is: shalom.malka.is@nps.edu.

Copyright Statement

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third-party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ICAS proceedings or as individual off-prints from the proceedings.

References

[1] Digital Twin Consortium, “Digital Twin,” Glossary. Available: https://www.digitaltwinconsortium.org/glossary/glossary.

[2] Gartner, “Gartner Identifies the Top 10 Strategic Technology Trends for 2018,” Oct. 2017. Available: https://www.gartner.com/en/newsroom/press-releases/2017-10-04-gartner-identifies-the-top-10-strategic-technology-trends-for-2018.

[3] AIAA Digital Engineering Integration Committee, “Digital Twin: Definition & Value, An AIAA and AIA

Position Paper," Dec. 2020. Available: https://aiaa.org/wp-content/uploads/2024/12/digital-twin-institute-position-paper-december-2020.pdf.

[4] Krill J. Systems Engineering of air and missile defenses," Johns Hopkins APL Technical Digest, Vol. 22, No. 3, pp. 220–233, 2001. Available: https://secwww.jhuapl.edu/techdigest/content/techdigest/pdf/V22-N03/22-03-Krill.pdf.

[5] Sommerer S, Guevara M, Landis M, Rizzuto J, Sheppard J, and Grant C. Systems-of-systems engineering in air and missile defense. Johns Hopkins APL Technical Digest, Vol. 31, No. 1, 2012. Available: https://secwww.jhuapl.edu/techdigest/content/techdigest/pdf/V31-N01/31-01-Sommerer.pdf.

[6] U.S. Army Space and Missile Defense Command, "Extended Air Defense Simulation (EADSIM)," Fact Sheet, 2022. Available: https://www.smdc.army.mil/Portals/38/Documents/Publications/Fact_Sheets/EADSIM.pdf

[7] Joint Air Power Competence Centre, "How can Modelling and Simulation Support Integrated Air and Missile Defence?" 2019. Available: https://www.japcc.org/articles/how-can-modelling-and-simulation-support-integrated-air-and-missile-defence.

[8] U.S. Department of Defense, "DoDI 5000.97: Digital Engineering," Dec. 21, 2023. Available: https://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodi/500097p.pdf.

[9] Office of the Deputy Assistant Secretary of Defense for Systems Engineering, "DoD Digital Engineering Strategy," Jun. 2018. Available: https://ac.cto.mil/wp-content/uploads/2019/06/2018-Digital-Engineering-Strategy_Approved_PrintVersion.pdf.

[10] White M. Adoption of digital twin within the Department of the Navy. Naval Postgraduate School, NPS-SE-22-263, 2022. Available: https://calhoun.nps.edu/bitstream/handle/10945/69516/NPS-SE-22-263.pdf.

[11] Zimmerman Z. Digital Engineering implementation across the Department of Defense. NDIA Systems and Mission Engineering Conference, Oct. 2019. Available: https://ndia.dtic.mil/wp-content/uploads/2019/systems/Wed_22447_Zimmerman.pdf.

[12] Hause M. A Systems engineering tools ecosystem. NDIA Systems Engineering Conference materials, 2018. Available: https://www.ndia.org/-/media/sites/ndia/divisions/systems-engineering/se-monthly-meetings/se---2018-2019-div-meeting-presentations/feb-2018/5_hause.pdf.

[13] Object Management Group, "OMG Systems Modeling Language (SysML) Version 1.6," formal/2019-11-01. Available: https://www.omg.org/spec/SysML/1.6.

[14] DoD Chief Information Officer, "DoD Architecture Framework (DoDAF) Version 2.02," Aug. 2010. Available: https://www.dmi-ida.org/knowledge-base-detail/The-DoDAF-Architecture-Framework-Version-202.

[15] U.S. Government Accountability Office, "Army Modernization: Air and Missile Defense Efforts Would Benefit from Applying Leading Practices," GAO-25-107491, Jun. 2025. Available: https://www.gao.gov/products/gao-25-107491.

[16] Northrop Grumman, "Integrated Battle Command System (IBCS)," product overview. Available: https://www.northropgrumman.com/what-we-do/missile-defense/battleone.

[17] Hodson D and Hill R. The art and science of live, virtual, and constructive simulation for test and analysis. Journal of Defense Modeling and Simulation, 2014. Available: https://journals.sagepub.com/doi/10.1177/1548512913506620.

[18] IEEE, "IEEE 1516 High Level Architecture (HLA) Standard," IEEE Standards Association. Available: https://standards.ieee.org/ieee/1516/6687.