INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY FUTURISTIC DEVELOPMENT

From Artificial Intelligence to Cyber-Physical Infrastructure Protection: A Unified Reliability and Security Engineering Framework

Sowmya Bodakunti
Master's in Computer Science, University of Houston Clear Lake, Houston, Texas, USA

* Corresponding Author: **Sowmya Bodakunti**

Article Info

P-ISSN: 3051-3618
E-ISSN: 3051-3626
Volume: 04
Issue: 01
January - June 2023
Received: 19-02-2023
Accepted: 21-03-2023
Published: 17-04-2023
Page No: 121-124

Abstract
Artificial intelligence is rapidly becoming embedded in software intensive and cyber physical infrastructure, where failures and attacks can propagate across digital and physical domains. Yet reliability engineering, safety engineering, and cybersecurity are still frequently practiced as parallel disciplines with different artifacts, metrics, and governance cadences. This paper proposes URSE, a Unified Reliability and Security Engineering framework that connects AI lifecycle controls, software reliability practices, and cyber physical infrastructure protection into one closed loop system. URSE introduces a shared system model, joint assurance artifacts, and a risk weighted reliability objective that aligns operational SRE metrics with infrastructure security outcomes. The framework is operationalized through decision intelligence loops that continuously translate telemetry into architecture level actions: control selection, test prioritization, automated rollback, and policy enforcement. We present a reference architecture and an evaluation design using a representative smart infrastructure scenario to show how URSE reduces mean time to recovery under combined fault plus attack conditions while improving measurable service availability and safety margin.

DOI: https://doi.org/10.54660/IJMFD.2023.4.1.121-124

Keywords: AI governance, cyber physical systems, infrastructure protection, reliability engineering, security engineering, decision intelligence, risk weighted availability

1. Introduction

Cyber physical infrastructure is increasingly software defined and AI assisted: forecasting, anomaly detection, scheduling, and automated control are now common in utilities, transportation, logistics, and healthcare adjacent systems. This integration increases capability but also creates coupled failure modes where a model error, a software regression, or an adversarial action can cascade into physical impact. Industrial control system environments further amplify this risk because of long system lifetimes, heterogeneity, and constrained patch windows. These characteristics are widely recognized as core drivers of cyber risk in operational technology contexts. [1]

Most organizations still treat reliability and security as separate delivery tracks. Reliability teams optimize uptime, latency, and incident response, while security teams optimize vulnerability reduction, control compliance, and threat response. In cyber physical environments, this separation is costly because safe operation is not purely a reliability property and secure operation is not purely a security property. The practical need is a unified engineering loop that can reason about both.

To establish a common language for AI risk and governance, URSE aligns AI controls to well defined risk functions and lifecycle requirements so that model development, deployment, monitoring, and change management remain traceable to operational outcomes. [9]

A second observation is that AI is already used in domains where safety, correctness, and accountability are primary concerns. Work on AI in clinical practice highlights the importance of controlled deployment, interpretability, and risk awareness when decisions affect people and operations. [2]

Goal of this paper: propose an actionable engineering framework that merges reliability, safety style hazard thinking, and cybersecurity into one operational system for AI enabled cyber physical infrastructure.

2. Motivation and Problem Definition

2.1. The coupled fault-attack space

In modern infrastructure stacks, the same event stream (telemetry) is used to drive both reliability responses (failover, rollback, autoscaling) and security responses (containment, segmentation, credential rotation). If these responses are not coordinated, they can conflict. For example, a reliability driven auto recovery may restore a compromised component, or a security driven containment may degrade safety constraints.

This paper models an infrastructure service as a set of interacting components: sensing, decision, control, and actuation. AI components may live in sensing (anomaly detection), decision (scheduling/optimization), or control (policy selection). Each layer has both failure modes and attack surfaces.

2.2. Unified objective

URSE defines a single objective: maximize mission availability subject to bounded risk. That requires a metric that can represent availability with security context without collapsing everything into compliance checklists.

3. Related Work and Foundations

3.1. Dependability and reliability foundations

URSE builds on established dependability concepts that distinguish faults, errors, and failures and emphasize that systems engineering must manage both internal failures and external disturbances. [6]

Classical software reliability engineering provides models and operational metrics (defect density trends, failure intensity, MTBF, MTTR) that remain relevant when AI is introduced, because AI systems still ship as software with deployment pipelines and runtime dependencies. [11]

3.2. Safety style hazard reasoning

Cyber physical incidents often emerge from unsafe control actions rather than a single component failure. URSE incorporates safety oriented system thinking to identify hazards, define constraints, and connect constraints to design and operational controls. [3]

3.3. Why AI governance must be operational

Governance that only exists in documentation does not protect live infrastructure. URSE treats governance as an executable loop: policy to telemetry to action, then feedback into architecture and model updates.

4. The URSE Framework

4.1. Core idea

URSE unifies four engineering planes into one closed loop:

1. System Model Plane: a shared model of components, dependencies, control boundaries, and failure/attack propagation paths.
2. Assurance Artifact Plane: joint artifacts that map hazards and threats to tests, controls, monitors, and recovery actions.
3. Operational Telemetry Plane: a single measurement fabric for reliability and security signals: traces, logs, metrics, config drift, policy drift.
4. Decision Intelligence Plane: a continuous planning loop that prioritizes engineering work using observed risk and reliability impact.

This decision intelligence approach is aligned with architecture centered lifecycle governance, where changes are evaluated for mission impact and driven through a measurable control loop. [5]

4.2. Risk weighted reliability metric

URSE uses a risk weighted availability concept to bind reliability and security outcomes:

Availability:

$$A = MTBF / (MTBF + MTTR)$$

Risk score R normalized to [0,[1] using threat exposure, control coverage, and anomaly indicators.

Risk Weighted Availability (RWA): $RWA = A \times (1 - R)$

The purpose is not to replace security metrics, but to create an optimization target that reliability and security teams can jointly improve without losing interpretability.

4.3. Joint assurance artifacts

URSE produces artifacts that are jointly owned:

Hazard-Threat Matrix: maps unsafe control actions and adversarial actions to the same impacted constraints.

Control Traceability Graph: links risks to controls, monitors, tests, and runbooks.

Change Impact Ledger: records what changed (model, config, code), what risks it can amplify, and what guards are required.

4.4. Lifecycle integration

URSE integrates into software lifecycle governance by using AI guided defect prediction, automated testing, and architectural checkpoints to prevent risk accumulation during rapid delivery cycles. [10]

As ML models take on broader responsibilities in software development and operations, governance must cover not only model behavior but also how models influence engineering decisions (for example, test selection or deployment approvals). [13]

5. Cyber-Physical Infrastructure Protection Mapping

5.1. Security scope in smart infrastructure

Infrastructure and public utilities require security controls that account for physical safety constraints, operational continuity, and regulatory expectations. These constraints differ from typical enterprise IT because downtime itself can be a safety hazard. [7]

A key gap in many programs is the fragmented boundary between information security and cybersecurity engineering in operational systems. URSE explicitly unifies these perspectives so that identity, network controls, and monitoring integrate with safety and reliability constraints rather than being bolted on late. [4]

Cyber physical environments also increasingly intersect with broader online safety concerns: exposure, misuse, and downstream impact are not limited to a single organization's perimeter. URSE treats this as part of risk propagation rather than an external topic. [12]

5.2. Control selection principle

URSE selects controls based on three questions:

1. Does the control reduce unsafe control action likelihood?
2. Does it reduce adversarial action likelihood or impact?
3. Does it reduce MTTR when prevention fails?

6. Methodology and Reference Architecture

6.1. Reference architecture components

URSE can be implemented using a layered architecture:

1. **Asset and dependency inventory:** service graph, OT boundaries, model registry.
2. **Unified telemetry pipeline:** metrics plus logs plus traces plus security events.
3. **Risk inference engine:** computes R from signals and policy state.
4. **Reliability engine:** computes A from incidents and SLO measurement.
5. **Decision intelligence orchestrator:** converts RWA gaps into actions (tests, controls, deployments, mitigations).
6. **Assurance repository:** stores traceability from risk to control to evidence.

6.2. Operational loop

1. **Observe:** collect telemetry and detect drift (data drift, config drift, control drift).
2. **Assess:** compute risk R, availability A, and RWA.
3. **Decide:** choose actions (patch, rollback, segmentation, test expansion, model retrain gating).
4. **Act:** execute changes through pipelines with enforcement gates.
5. **Learn:** update hazard-threat mappings and governance rules.

6.3. Economics of automation

URSE includes explicit measurement of the cost and time impact of automation because infrastructure programs often fail when governance overhead slows delivery. Automation ROI framing supports prioritization of controls and tests that deliver measurable savings while improving assurance. [8]

7. Evaluation Design and Illustrative Scenario

7.1. Scenario

We evaluate URSE using a representative smart infrastructure service (for example, a utility control support system) where an AI model detects anomalies and recommends control actions, a microservice layer executes scheduling, policy checks, and command staging, and OT boundary systems apply control actions under safety constraints. The evaluation considers combined events: (i) software regression causing intermittent command delays, and (ii) an adversarial attempt to manipulate telemetry signals.

7.2. Metrics

Reliability: MTTR, incident frequency, availability A. Security: control coverage index, anomaly persistence, risk score R.

Unified: risk weighted availability (RWA). Governance: time from detection to enforced mitigation, and percentage of changes with complete traceability.

7.3. Expected outcome pattern

URSE's value is demonstrated when it reduces time to safe containment while preserving mission continuity. The framework is designed so that security containment actions are constraint aware (do not violate safety) and reliability recovery actions are compromise aware (do not restore a known bad state). This is most visible under mixed fault plus attack workloads in which single discipline responses often conflict.

8. Discussion

8.1. Practical adoption

URSE is intentionally compatible with existing practices: SLOs, incident management, threat modeling, and governance reviews. The main change is forcing convergence at the artifact level and at the operational decision level.

8.2. Where URSE is strongest

URSE is strongest in systems where AI outputs influence operational actions, in environments where downtime is safety relevant, and in organizations struggling with fragmented reliability versus security ownership.

8.3. Limitations

URSE requires sufficient telemetry maturity and a trustworthy dependency graph. In very legacy OT environments, data collection constraints may limit automated risk inference, and some controls will remain manual until instrumentation improves.

9. Conclusion

AI enabled cyber physical infrastructure demands engineering approaches that treat reliability, safety constraints, and cybersecurity as one coupled problem. URSE provides a unified framework with a shared system model, joint assurance artifacts, and an operational decision intelligence loop guided by risk weighted reliability. By aligning AI governance with runtime enforcement and engineering economics, URSE supports measurable improvements in both continuity and protection while remaining practical for real delivery pipelines.

References

1. National Institute of Standards and Technology (NIST). Guide to Industrial Control Systems (ICS) Security. NIST Special Publication 800-82 Rev. 2; 2015.
2. Kacheru G. Revolutionizing healthcare: the role of artificial intelligence in clinical practice. J Comput Anal Appl. 2023;31(4):1544–1546. Available from: https://eudoxuspress.com/index.php/pub/article/view/3270
3. Leveson NG. Engineering a safer world: systems thinking applied to safety. Cambridge (MA): MIT Press; 2011.
4. Pittala SK, Ashok VKC. A new era in security: bridging information security and cybersecurity. Int J Multidiscip Futuristic Dev. 2023;4(1):69–72. doi:10.54660/IJMFD.2023.4.1.69-72
5. Gunda SK, Yettapu SDR, Bodakunti S, Bikki SB. Decision intelligence methodology for AI-driven agile software lifecycle governance and architecture-centered

project management. 2023;4(1):102–108. Available from: https://doi.org/10.63282/3050-9262.IJAIDSML-V4I1P112
6. Laprie JC. Dependable computing and fault tolerance: concepts and terminology. In: Proceedings of the 25th International Symposium on Fault-Tolerant Computing (FTCS-25); 1995.
7. Ashok VKC. Cybersecurity for smart infrastructure and public utilities. Int J Multidiscip Res Growth Eval. 2023;4(2):947–949. doi:10.54660/.IJMRGE.2023.4.2.947-949
8. Kacheru G, Bajjuru R, Arthan N. The ROI of software automation: measuring time and cost savings. Int J Commun Netw Inf Secur. 2023;15(4):774–785.
9. National Institute of Standards and Technology (NIST). Artificial Intelligence Risk Management Framework (AI RMF 1.0). NIST AI 100-1; 2023.
10. Sivva SD, Thalakanti RR, Bandari SSG, Yettapu SDR. AI-driven decision intelligence for agile software lifecycle governance: an architecture-centered framework integrating machine learning defect prediction and automated testing. 2023;4(4):167–172. Available from: https://www.ijetcsit.org/index.php/ijetcsit/article/view/554
11. Lyu MR, editor. Handbook of software reliability engineering. New York: McGraw-Hill; 1996.
12. Pittala SK. Cybersecurity and online safety: a critical asset in the information era. J Front Multidiscip Res. 2023;4(1):576–579. doi:10.54660/.jfmr.2023.4.1.576-579
13. Gunda SKG. The future of software development and the expanding role of ML models. Int J Emerg Res Eng Technol. 2023;4(2):126–129. Available from: https://doi.org/10.63282/3050-922X.IJERET-V4I2P113
14. International Electrotechnical Commission (IEC). IEC 61508: Functional safety of electrical/electronic/programmable electronic safety-related systems. 2010.
15. International Organization for Standardization/International Electrotechnical Commission (ISO/IEC). ISO/IEC 27001:2013 Information technology – security techniques – information security management systems – requirements. 2013.
16. National Institute of Standards and Technology (NIST). Security and privacy controls for information systems and organizations. NIST SP 800-53 Rev. 5; 2020.
17. MITRE Corporation. ATT&CK for ICS (Industrial Control Systems) knowledge base. 2020.
18. Sculley D, Holt G, Golovin D, *et al*. Hidden technical debt in machine learning systems. In: Advances in Neural Information Processing Systems (NeurIPS); 2015.
19. Lee EA, Seshia SA. Introduction to embedded systems: a cyber-physical systems approach. 2nd ed.; 2017.
20. Kriaa S, Bouissou M, Pietre-Cambacedes L. A survey of approaches combining safety and security for industrial control systems. Reliab Eng Syst Saf. 2015.
21. Humayed A, Lin J, Li F, Luo B. Cyber-physical systems security: a survey. IEEE Internet Things J. 2017.
22. Langner R. Stuxnet: dissecting a cyberwarfare weapon. IEEE Secur Priv. 2011.