Proc. of the European Safety and Reliability Conference (ESREL 2026)
Edited by José C. Matos, Paulo B. Lourenço, Michael Beer, Daniel Oliveira, Edoardo Patelli, Hélder S. Sousa, Rui A. Silva and Monica Santamaria
 Published by Research Publishing, Singapore.
doi: 10.3850/ESREL2026061419_esrel26-p27687-cd

Managing Change in Maritime AI: A Semantic and Modular Framework for Adaptive Maritime Systems

Nikita Karandikar

Research, DNV, Norway. E-mail: nikita.karandikar@dnv.com

Shuai Wang

Research, DNV, Norway. E-mail: shuai.wang@dnv.com

Knut Erik Knutsen

Research, DNV, Norway. E-mail: knut.erik.knutsen@dnv.com

Artificial intelligence (AI) is transforming maritime operations with increasingly autonomous functions. These systems must adapt continuously to changing environmental conditions, new technologies, diverse fleets, and evolving regulations to maintain reliability, safety, and compliance which are critical factors in the maritime industry. This paper presents a semantic and modular framework designed to manage such change through interconnected layers. The base layer employs a maritime knowledge graph based on ISO 19848 Annex C to integrate operational data from sensors and navigation systems. Above this, a layer of modular contracts wraps AI models and system modules, allowing easy updates and verification as conditions evolve. Modular contracts represent formal agreements specifying requirements, acceptance criteria, and interfaces for modules, such as an autonomous navigation system. These contracts define module interactions, enabling reuse across ships and simplifying updates and verification. A federated learning layer supports secure AI training and updates distributed across multiple vessels. An explainable AI layer provides operator insights to foster transparency and trust. The top layer applies dynamic risk management and compliance checks aligned with regulations. An illustrative example of ship propulsion fault detection shows how these layers interoperate—from data integration and modular design to evolving AI, human oversight, and risk management, offering a flexible approach to managing change in maritime AI systems. This framework provides an adaptable and robust foundation for AI solutions that must remain effective and safe in the ever-changing maritime environment.

Keywords: maritime AI, knowledge graph, modular contracts, risk, change management, assurance, reliability.

1. Introduction

Autonomous maritime cyber-physical systems are evolving to incorporate increasingly sophisticated autonomous functions such as navigation, propulsion control, safety monitoring, and communications. This growth significantly raises system complexity due to heightened interactions among subsystems and the emergence of new fault modes and systemic hazards, challenging traditional fault analysis and assurance frameworks. Ensuring safe, secure, and resilient operation in such dynamic environments requires an integrated approach combining rigorous hazard identification, detailed data-driven monitoring, explainable fault diagnosis, and effective change management.

System-Theoretic Process Analysis (STPA), including its security extensions, is a well-established top-down methodology for hazard identification in complex cyber-physical systems. By systematically analyzing unsafe control actions and enforcing safety and security constraints, STPA offers a structured framework to understand hazard propagation and systemic risk in autonomous maritime contexts. In contrast, semantic knowledge graphs represent an emerging bottom-up technology modeling detailed system components, interconnections, operational data, and domain context. Combined with federated learning and explainable graph-based AI, these models support privacy-preserving fault detection, causal reasoning, and interpretable diagnostics across distributed fleets.

This paper proposes a layered architecture integrating the established capabilities of top-down STPA with the emerging semantic knowledge graph approach. High-level hazards and control constraints identified through STPA are anchored to concrete entities and relationships within the knowledge graph, enabling bidirectional causal reasoning that connects abstract hazard models with observed operational data. The Vessel Information Structure (VIS) (DNV, 2026) based on the ISO 19848 Annex C standard provide essential semantic frameworks to ensure interoperability, consistent data exchange, and integration with shipboard ICT systems. These standards underpin the semantic layer, facilitating alignment of assurance evidence with digital representations of vessels and their subsystems.

Modular assurance is employed through contract-based designs encapsulating autonomous subsystems with clearly defined assumptions, guarantees, and interfaces. This modularity supports scalable assurance composition and allows localized updates and verification. Change management is integral to the framework, ensuring that as autonomous functions evolve through software updates or reconfigurations, hazard models, semantic representations, AI diagnostics, and contractual assurances remain synchronized to maintain valid fault analysis and trustworthy system operation.

A propulsion fault detection use case illustrates how this integrated framework combines top-down hazard analysis, bottom-up semantic and AI-powered reasoning, modular contract assurance, and systematic change management to provide scalable, explainable, and privacy-aware fault diagnosis and assurance tailored to autonomous maritime cyber-physical systems.

2. Background and Related Work

Autonomous maritime systems are rapidly transforming the shipping industry, introducing notable safety and security challenges due to their increasing complexity and novel operational modes. Glomsrud and Xie (2019) show that STPA and its security extension STPA Sec effectively identify hazards through unsafe control actions, making them well suited to autonomous maritime environments. Semantic knowledge graphs support maritime cyber physical systems by integrating data and enabling reasoning over assets, sensors, software, and operations. Cybersecurity knowledge graphs model maritime specific threats and attack paths, supporting threat hunting and cyber risk assessment, and have been used to detect multifaceted cyber attack scenarios (Sikos, 2023).

Federated learning adapted to maritime domains allows privacy preserving, collaborative model training over distributed vessel data (Zhang et al., 2021). Explainable AI methods based on Graph Neural Networks, such as GNNExplainer (Ying et al., 2019), highlight influential subgraphs behind anomalies, improving operator understanding and trust.

Further, contract based governance frameworks specify assumptions, guarantees, and interfaces, enabling systematic verification and safe evolutionary updates of autonomous subsystems (Glomsrud et al., 2024). Despite these advances, integrating STPA/STPA Sec with semantic knowledge graphs, federated learning, XAI, cybersecurity knowledge graphs, and contract based assurance into a coherent, scalable architecture remains an open challenge, especially when maintaining assurance through lifecycle change.

Our paper addresses this by proposing an integrated layered architecture that combines these methods to deliver scalable, explainable, and privacy respecting fault analysis and risk management for autonomous maritime cyber physical systems.

3. Framework Architecture and Methodology

3.1. *Layered Modular Design*

The framework adopts a modular, layered architecture, as shown in Fig. 1, to manage complexity in autonomous maritime systems by separating concerns and enabling scalable, adaptable assurance.

3.1.1. *Knowledge Graph Layer:*

This semantic foundation constructs a rich, interconnected graph model of the ship's physical subsystems, sensors, operational environment, and their relationships. Leveraging maritime ontologies from VIS, it accurately captures relevant maritime domain concepts. The knowledge graph integrates heterogeneous data sources into a unified, semantically annotated representation that transforms raw sensor streams and system logs into contextualized triples, enabling advanced querying and reasoning essential for fault cause detection and system dependency analysis.

3.1.2. ***Federated Learning Layer:***

Addressing the distributed nature of maritime fleets and stringent data confidentiality, federated learning algorithms train anomaly detection models locally on each vessel's enriched semantic data. Periodic aggregation of model updates creates a robust global model while maintaining data decentralization, thus respecting maritime communication bandwidth and privacy constraints. This layer supports continuous learning and adapts to diverse operational profiles across varied environments.

3.1.3. ***Explainability Layer - GNNExplainer:***

Graph Neural Networks (GNNs) form the AI core by leveraging relationships encoded in the knowledge graph for fault prediction and anomaly detection. GNNExplainer renders these predictions interpretable by identifying minimal subgraphs and node features significantly influencing each output. It solves an information-theoretic optimization to extract subgraphs that locally approximate the GNN's decision boundaries, producing concise, faithful explanations tied to actual system components and interactions. This transparency builds operator trust and aids diagnosis by revealing root causes of faults.

3.1.4. ***Cybersecurity Knowledge Graph Layer:***

Cyber-physical security threats are modeled in a dedicated, evolving knowledge graph capturing known vulnerabilities, attacker behaviors, and attack paths specific to maritime autonomous systems. By correlating cybersecurity data with operational states, this layer supports real-time anomaly detection and cyber-risk assessment, protecting critical ship functionalities.

3.1.5. ***Contract Governance Layer - Assume-Guarantee Contracts:***

Modular assurance relies on assume-guarantee contracts that assign explicit behavioral assumptions and guarantees to each subsystem. Verifying correctness under these contracts enables compositional, incremental assurance. Changes such as software updates, trigger reevaluation only of affected contracts and their neighbors. This modular verification is crucial for managing evolving autonomous systems while ensuring regulatory compliance.

3.1.6. ***Operator Risk Dashboard*:**

All layers feed an actionable operator interface aggregating fault predictions, explanation subgraphs, cybersecurity alerts, contract statuses, and overall risk assessments. Human operators retain supervisory control, receive understandable explanations, and intervene dynamically as needed.

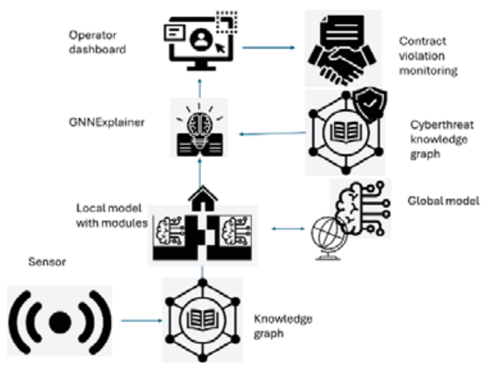

Fig. 1. Framework Architecture

3.2. ***Layer Interactions and Data Flow***

The architecture supports dynamic, bidirectional information flow by semantically integrating raw sensor and system data into the knowledge graph according to maritime standards, enabling federated learning to train local anomaly-detection models on this semantic data with periodic global aggregation. GNN-based inference then operates on relevant subgraphs to generate predictions, while GNNExplainer extracts causal subgraphs to clarify the origins of detected anomalies. In parallel, cybersecurity graph analysis cross-references threat patterns with operational states to identify potential risks, and real-time knowledge-graph data is continuously verified against assume-guarantee contracts to ensure compliance. These outputs—predictions, explanations, cyber alerts, and contract results—are synthesized into composite risk assessments, which are presented to operators through intuitive dashboards. Operator inputs in turn feed adaptive loops that trigger model retraining, contract refinement, or operational adjustments, ensuring continuously evolving and trustworthy system assurance.

3.3. *Technical Workflow and Pseudocode*

This subsection delineates and describes the framework workflow through pseudocode of the main components.

```
# Initialize semantic knowledge graph with maritime ontologies
knowledge_graph = initialize_knowledge_graph(ISO19848_ontology)
# Load modular assume-guarantee contracts for ship subsystems
contracts = load_contracts()
# Initialize cybersecurity knowledge graph for threat modeling
cybersecurity_kg = initialize_cybersecurity_graph()
# Initialize federated global anomaly detection model
global_model = initialize_federated_model()
while system_is_operational():

  # Step 1: Data collection and semantic integration per vessel
  for vessel in fleet:
    raw_data = collect_real_time_data(vessel)
    semantic_data = annotate_data(raw_data, knowledge_graph.schema)
    knowledge_graph.update(vessel, semantic_data)

  # Step 2: Local federated learning on semantic data at each vessel
  for vessel in fleet:
    vessel_data = knowledge_graph.query(vessel)
    local_update = train_local_anomaly_detector(global_model, vessel_data)
    send_local_update_to_server(vessel, local_update)

  # Step 3: Aggregate updates from all vessels centrally
  fleet_local_updates = collect_all_local_updates(fleet)
  global_model = aggregate_updates(fleet_local_updates)
distribute_global_model_to_vessels(global_model)

  # Step 4: Anomaly prediction and explanation on each vessel
  for vessel in fleet:
    reasoning_subgraph = knowledge_graph.extract_relevant_subgraph(vessel)
    anomaly_prediction = global_model.predict(reasoning_subgraph)
    if anomaly_prediction:
      explainer = GNNExplainer(global_model, reasoning_subgraph)
      explanation_subgraph = explainer.explain()

knowledge_graph.annotate_explanation(vessel, explanation_subgraph)
      cyber_alerts = cybersecurity_kg.analyze(vessel, knowledge_graph.get_state(vessel))
```

```
        contract_status = verify_contracts(knowledge_graph, contracts, vessel)
        risk_level = synthesize_risk(anomaly_prediction, cyber_alerts, contract_status)
        operator_dashboard.update(vessel, risk_level, explanation_subgraph, cyber_alerts)

 # Step 5: Operator feedback loop for adaptive assurance
        feedback = operator_dashboard.retrieve_feedback(vessel)
        if feedback:
          update_global_model(global_model, feedback)
          revise_contracts(contracts, feedback)
knowledge_graph.update_with_feedback(vessel, feedback)
# GNNExplainer algorithm (simplified)
class GNNExplainer:
  def __init__(self, model, subgraph):
    self.model = model
    self.subgraph = subgraph
  def explain(self):
    mask = initialize_mask(self.subgraph)
    for epoch in training_epochs:
        pred_full = self.model.predict(self.subgraph)
        pred_masked = self.model.predict(apply_mask(self.subgraph, mask))
        loss = -mutual_information(pred_full, pred_masked) + regularization(mask)
        mask.grad = compute_gradient(loss, mask)
        mask = update_mask(mask, mask.grad)
    explanation_subgraph = extract_masked_subgraph(self.subgraph, mask)
    return explanation_subgraph
# Verify modular contracts for compliance
def verify_contracts(knowledge_graph, contracts, vessel):
  related_contracts = contracts.get_for_vessel(vessel)
  for contract in related_contracts:
    if not contract.is_satisfied(knowledge_graph, vessel):
        return "violation"
  return "satisfied"
```

3.4. *GNNExplainer Intuition*

GNNExplainer begins by considering all nodes and edges in the subgraph as potentially relevant to the model's prediction, then constructs a mask that acts as a filter to determine which components to retain or suppress. It evaluates the GNN's prediction on both the full subgraph and the masked version to assess how well the mask preserves the original outcome, calculating the difference between these results to gauge fidelity. The mask is then iteratively refined, step by step, to retain only the most influential nodes and edges while ensuring the prediction remains effectively unchanged. This refinement loop continues until the algorithm converges on the smallest subgraph capable of reproducing the original prediction, which it returns as the explanation—the minimal causal structure driving the GNN's decision. This mechanism integrates seamlessly with the broader assurance framework, which unifies semantic data representation, distributed collaborative learning, explainable AI reasoning, cybersecurity monitoring, formal verification through assume-guarantee contracts, and operator-driven feedback loops. Together, these elements form a continuous, closed-loop assurance process tailored to maritime systems' needs for fault detection, intelligible explanations, security, compliance, and adaptive operation.

4. Application Example: Adaptive Assurance in Ship Propulsion Shaft Fault Management

The propulsion shaft is a critical mechanical component transferring power from engine to propeller, subject to cyclic mechanical loads including torsional vibrations, bending moments, axial thrust, and environmental forces such as wave action. These loads induce fatigue failures that concentrate at geometric stress raisers like fillets, tapers, chamfers, and welds. Material imperfections, corrosion, and overload further exacerbate fatigue crack initiation and propagation, threatening structural integrity and operational safety. System-Theoretic Process Analysis (STPA) systematically maps unsafe control actions such as dwelling too long in barred speed ranges or insufficient vibration monitoring to hazardous conditions like catastrophic shaft fractures or loss of propulsion. These causal pathways form the foundation for hazard-informed system requirements. Modern fatigue hazard management also incorporates cyber-physical concerns: sensor spoofing, denial-of-service, or data corruption risks that can mask early fault detection. The semantic knowledge graph conforms to ISO 19848, standardizing sensor and machinery representation, enabling precise fault localization and causal inference.

4.1. *Case 1: Coupling Bolt Fatigue (Complex Fault Needing Explainability and Override)*

4.1.1. *Phenomenon*

Coupling bolts joining shaft segments are vulnerable to fatigue from dynamic loads—torque overloads, thrust reversals during astern maneuvers, bending moments, and wave-induced vibrations, especially in older vessels. Fatigue fractures of these bolts can result in immediate loss of propulsion and maneuverability.

4.1.2. *System Detection*

The ISO 19848-compliant semantic knowledge graph identifies sensors (strain gauges, accelerometers) located on shaft collars and coupling bolts whose anomalous readings correlate with fatigue risk. Federated graph neural networks (GNNs) predict high anomaly probabilities with GNNExplainer methods localizing influential sensors and physical components. Minimal influential subgraphs in a GNN context mean identifying the smallest pieces of the semantic graph in this case key sensors and machine parts, that most strongly affect the model's prediction. This helps us understand why the model predicted a fault by focusing on critical elements in the graph.

```
def identify_and_explain_coupling_bolt_fault(kg):
    sensors = kg.query("strain_gauges_on_coupling_bolts")
    critical_sensor = kg.select_most_correlated_sensor(sensors)
    subgraph = kg.get_subgraph(critical_sensor)
    anomaly_prob = gnn_model.predict(subgraph)
    explanation = gnn_explainer.explain(gnn_model, subgraph)
    operator_interface.alert(f"Coupling bolt fatigue risk detected at sensor {critical_sensor}", explanation)
```

4.1.3. *Operator override justification*

An operator, understanding that immediate propulsion shutdown during a critical docking maneuver might cause hazardous drift or collision, overrides automatic fault mitigation commands, opting for continued operation and scheduled repair.

```
def process_operator_override(vessel_state, override_flag):
    if override_flag:
        vessel_state['override_fault_mitigation'] = True
        vessel_state['schedule_repair'] = True
        log_event("Operator override: Deferred immediate propulsion shutdown for docking safety")
    return vessel_state
```

4.1.4. *Knowledge Graph Update*

The override feedback is encoded in the semantic model, updating hazard propagation and mitigation assumptions for continued risk assessment.

```
def update_kg_with_override_feedback(kg, vessel_state):
    kg.add_fact('override_fault_mitigation', vessel_state['override_fault_mitigation'])
    kg.update_hazard_assumptions()
```

4.2. *Case 2: Sensor-Detected Emerging Fatigue Fault (Bottom-Up Federated Learning)*

In this bottom-up scenario, the Sensor Layer first collects ISO 19848-compliant vibration, temperature, and strain data streams, which the Diagnostic Layer processes using local GNNs that analyze semantic sensor embeddings to detect emerging fatigue-related anomalies. These local insights feed into the Federated Learning Layer, where distributed model updates are aggregated into a global GNN to enhance fleet-wide diagnostic accuracy. The Explainability Layer then applies GNNExplainer to isolate the minimal influential subgraphs underlying each anomaly prediction, providing transparent, causally grounded explanations. These results are presented through the Operator Interface Layer, which visualizes fault probabilities alongside their explanations to support informed decision-making. Finally, the Contract and Monitoring Layer assimilates newly identified fault scenarios into formal assume-guarantee contracts and continuously checks runtime behavior for compliance, ensuring adaptive, standards-aligned assurance.

```
def bottom_up_fault_detection(vessel_data, gnn_local, federated_server, kg, operator_dashboard):
    semantic_data = kg.embed(vessel_data.sensor_streams)
    local_update = gnn_local.train(semantic_data)
    federated_server.submit_update(local_update)
    global_model = federated_server.aggregate()
    subgraph = kg.extract_relevant_subgraph(vessel_data.id)
    fault_prob = global_model.predict(subgraph)
    explanation = GNNExplainer(global_model, subgraph).explain()
    operator_dashboard.update(vessel_data.id, fault_prob, explanation)
    contracts.adapt(global_model)
    violations = contracts.verify(vessel_data.system_state)
    if violations:
        operator_dashboard.alert(violations)
        mitigation_protocols.execute(violations)
```

4.3. *Case 3: Software Update and Contract-Based Assurance*

In the software-update scenario, the Software Management Layer first applies the patch and adjusts relevant model parameters, prompting the Diagnostic Layer to retrain the GNN-based detection model to ensure it reflects the updated system behavior. The Contract Layer then adapts the assurance contracts to incorporate any new assumptions or guarantees introduced by the update and performs compliance checks to validate correctness. These contracts are continuously overseen by the Runtime Verification Layer, which monitors adherence during operation and triggers alerts or mitigation actions when deviations occur. Throughout this process, the Operator Interface Layer keeps operators informed by presenting contract statuses, runtime warnings, and any required operational responses, ensuring transparency and trustworthy post-update assurance.

```
def software_update_assurance(global_model, patch, vessel_state, contracts, operator_interface):
    global_model.apply_patch(patch)
    global_model.retrain()
    updated_contracts = contracts.adapt(patch)
    verification_results = contracts.verify(updated_contracts, vessel_state)
    if verification_results.has_violations():
        operator_interface.alert(verification_results.get_violation_details())
        mitigation_protocols.execute(verification_results.get_violations())
```

5. Discussion and Conclusion

This integrated framework presents a comprehensive approach to maritime cyber-physical system (CPS) safety and security, addressing mechanical fatigue detection, cybersecurity risks, and human factors in propulsion shaft fault management.

The framework's layered design, spanning ISO 19848-compliant semantic knowledge graphs, federated AI diagnostics with explainability, cyber-physical risk fusion, formal runtime contract verification, and operator-centered interfaces delivers several core benefits. It effectively mitigates emerging cyber-physical threats while enhancing operator trust through interpretable AI, supports modular and scalable assurance aligned with maritime standards, and adapts dynamically to changing operational contexts via human feedback.

However, significant challenges remain. Computational limitations on onboard AI training and communication impose strict efficiency demands. The evolving threat landscape requires continuous advancement in cybersecurity analytics and AI diagnostics. System complexity mandates rigorous integration and validation practices, and human factors challenge balanced and effective interpretation of AI outputs in operational decisions.

Emerging technologies including edge computing, adaptive federated learning, augmented reality-enhanced operator aids, and advanced knowledge graph cyber analytics signal promising pathways to overcome these challenges and further fortify maritime CPS assurance.

In conclusion, this work lays a robust foundation for next-generation autonomous maritime systems that are safer, more transparent, and resilient. It highlights critical interdependencies between mechanical safety, cyber defenses, and human oversight that must inform future maritime digitalization and regulatory frameworks, ensuring sustainable and reliable seafaring in rapidly evolving technological landscapes.

Acknowledgement

This work is supported by the WARRaNT project (no. 101202581) funded by the European Commission.

References

DNV. “VISTA.” Accessed February 6, 2026. https://vista.dnv.com/.

Glomsrud, Jon Arne, and J. Xie. "A structured STPA safety and security co-analysis framework for autonomous ships." In European Safety and Reliability conference, Germany, Hannover. 2019.

Sikos, Leslie F. "Cybersecurity knowledge graphs." *Knowledge and Information Systems* 65, no. 9 (2023): 3511-3531.

Zhang, Zehui, Cong Guan, Hui Chen, Xiangguo Yang, Wenfeng Gong, and Ansheng Yang. "Adaptive privacy-preserving federated learning for fault diagnosis in internet of ships." *IEEE Internet of Things Journal* 9, no. 9 (2021): 6844-6854.

Ying, Zhitao, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. "Gnnexplainer: Generating explanations for graph neural networks." *Advances in neural information processing systems* 32 (2019).

Arne Glomsrud, Jon, Stephanie Kemna, Chanjei Vasanthan, Luman Zhao, Dag McGeorge, Tom Arne Pedersen, Tobias Rye Torben, Børge Rokseth, and Dong Trong Nguyen. "Modular assurance of an autonomous ferry using contract-based design and simulation-based verification principles." In *Journal of Physics: Conference Series*, vol. 2867, no. 1, p. 012043. IOP Publishing, 2024.