An MBSE Driven Framework for Automated Generation of RAM Risk Models from Maritime Vessel System Architectures

Yanlai Wang[1], Toby Adam Michael Russell[2], and Octavian Niculita[3]

[1,2]*Ocean Infinity Ltd, Southampton, Hampshire, SO19 9UY, UK*
yanlai.wang@oceaninfinity.com
toby.russell@oceaninfinity.com

[2,3]*Glasgow Caledonian University, Glasgow, Scotland, G4 0BA, UK*
octavian.niculita@gcu.ac.uk

ABSTRACT

Reliability, availability, and maintainability (RAM) analysis in maritime systems depends on risk models that are typically developed manually from design documentation. This process is time-consuming, error-prone, and weakly connected to system architecture models. This work presents an MBSE-driven framework that transforms SysML system models into RAAML-based risk models, establishing a direct link between system design and RAM analysis.

The framework defines a multi-layer mapping approach covering concept, semantic, and meta-model levels. Structural, behavioural, and interface elements in SysML are systematically converted into risk modelling constructs, enabling the automatic generation of reliability block diagrams, mission profiles, and failure propagation models. Quantitative parameters, including failure rates and maintenance data, are incorporated to support system-level reliability evaluation. The derived propagation paths and critical dependencies also provide a basis for sensor placement and condition monitoring design, supporting PHM applications.

A maritime case study based on a remotely operated vehicle (ROV) demonstrates the applicability of the approach. The results show a reduction in RAM modelling effort of 34% to 55% compared to conventional methods, with greater benefits observed for more complex systems. The framework improves modelling consistency, traceability, and scalability by deriving risk models directly from system architecture.

1. OVERVIEW

The increasing complexity of maritime engineering systems has driven the adoption of Model-Based Systems Engineering (MBSE) using SysML to support system development and maintenance across the system lifecycle. At the same time, Reliability, Availability, and Maintainability (RAM) analysis plays a critical role in evaluating system risk and operational performance.

Several studies have attempted to integrate system architecture modeling with reliability analysis. Biggs, Geoffrey & Juknevicius, Tomas & Armonas, Andrius & Post, Kyle. (2018) proposed extending SysML to incorporate safety and reliability analysis concepts within the MBSE workflow. Similarly, Diatte, K.; O'Halloran, B.; Van Bossuyt, D.L.(2022) discussed integrating RAM analysis into MBSE process to improve design decision-making. In the maritime domain, V. Sideris, Z. P. Oikonomou, S. Gerené, and A. A. Kana (2025) demonstrated the MBSE methodology for naval vessel system architecture design, highlighting the benefits of model-based representation for complex ship systems, although their work did not address reliability or risk modeling aspects.

To support model-based risk analysis, the Risk Analysis and Assessment Modeling Language (RAAML) was introduced by the Object Management Group (OMG) in 2016. RAAML extends SysML concepts by providing standardized constructions for modeling risk analysis such as fault trees, failure modes, and reliability structures. Hecht, M., Juknevičius, T., Post, K. and Armonas, A. (2024/2025) built the extend principles from SysML to RAAML by adding Reliability Block Diagram (RBD).

In the aerospace domain, B. Kaiser, M. Soden and N. Heuermann, (2024) demonstrated a UAV case study in which Component Fault Trees (CFT) was automatically generated from SysML models. E. Brusa, (2021) integrated RAM analysis into aircraft system design workflows and generated FMEA models from system architecture models.

Related work has also explored Model-Based Safety Analysis (MBSA) approaches. Girard, Gaëlle & Baeriswyl, Ivan & Hendriks, Jonathan & Scherwey, Roland & Müller, Christian & Hönig, Philipp & Lunde, Rüdiger. (2020)

---

proposed an MBSA workflow that automatically generates FMEA and fault tree models from SysML using the smartIflow tool. W. Huang and D. Borja (2025) presented a reliability assessment approach combining MBSE and FMEA, where system reliability is evaluated using RBD derived from system models. M. W. Daniels and K. Pierre, (2023) illustrated MBSE-based reliability analysis workflow using both spreadsheet-based and model-based implementations.

In addition, probabilistic extensions to MBSE tools have been investigated. Cvijic S., Harrison S., Marotta S. (2026) and Cvijic S., Hookway S., Harrison S.and Strelzoff A., (2025) proposed a plugin for Cameo Systems Modeler that enables reliability analysis using the ProbSysML framework to support probabilistic simulation of systems-of-systems. Suls J., (2024) introduced mathematical algorithms to evaluate RAM properties directly on system models developed in Cameo. Industrial tools such as the Maintenance Aware Design Environment (MADe™) also support risk-driven design by integrating system modeling with reliability analysis capabilities. Niculita, Ioan-Octavian; Jennions, Ian K.; Medina Valdez, Miguel. (2014) investigated the automatic generation from CAD models to risk models which can accelerate risk modelling workflows and reduce modelling effort. However many of these approaches rely on specific tools or customized model extensions, which limit their interoperability across different MBSE environments.

Overall, most existing approaches focus on either extending SysML to represent reliability concepts or embedding reliability analysis within specific MBSE tools. A significant gap therefore remains between system models and risk models. Therefore, a systematic transformation framework which can derive RAM risk models directly from SysML-based system architecture models remains an open research challenge.

2. GENERAL PRINCIPLES

Unlike existing approaches that extend SysML or rely on tool-specific integration, a language-level transformation between SysML and RAAML is defined to enable tool-independent model generation.

The systems engineering V-model is widely used to organize system development and verification activities throughout the system lifecycle. In the traditional V-model, the left side represents system definition and design activities, while the right side represents integration, verification, and validation processes. System requirements are progressively refined into system architecture, detailed design, and implementation, followed by corresponding verification and testing activities. However, in many traditional development processes, reliability modelling and PHM design are often introduced only after major design decisions have been made. This separation limits the traceability between system architecture and risk models and makes it difficult to ensure consistency between design and reliability analysis. To address this limitation, the framework extends the V-model by integrating RAM modelling and PHM design directly into the system engineering process. Figure 1 illustrates the updated V-model used to explain the general principle of the framework.

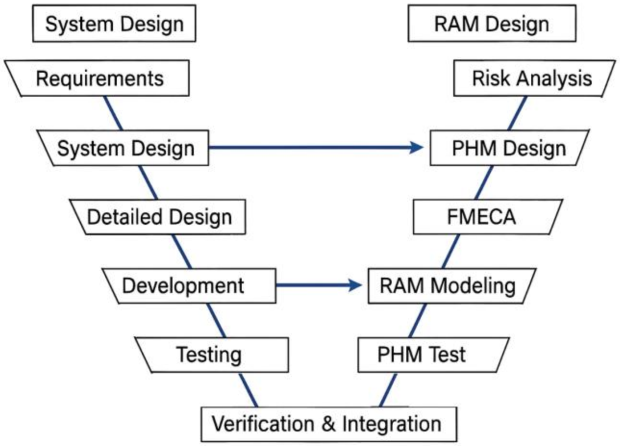


Figure 1. Update V-model demonstrating the integration of RAM activities

As shown in Figure 1, RAM modelling is derived directly from system development results, while PHM design is also generated based on system architecture and design information. In this updated V-model, RAM analysis and PHM design are closely linked with system design activities, enabling reliability considerations to be incorporated earlier in the development process.

Figure 2 presents the detailed structure of the extended model. In the framework, RAM analysis is derived from the system architecture during the early design stages, allowing reliability considerations to influence detailed design, sensor design, and implementation decisions while maintaining consistency with verification and testing activities.

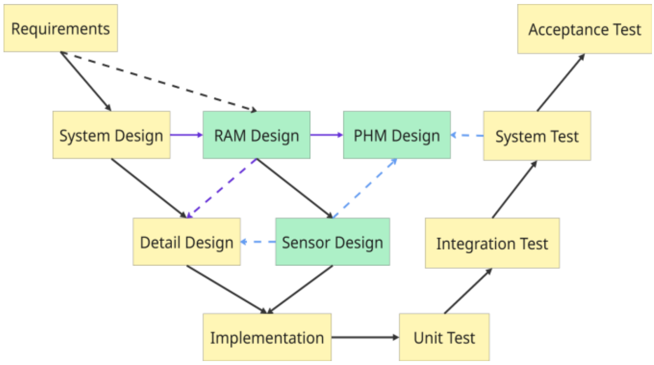


Figure 2. Extended V-model integrating RAM and PHM design

The extension of the V-model provides several advantages. First, it ensures that RAM analysis is based on authoritative sources of truth, improving the consistency and accuracy of the resulting risk models. Second, it reduces modelling effort and cost by deriving functional and reliability models directly from system models rather than reconstructing them from design documentation, thereby minimizing inconsistencies and improving model fidelity. Finally, and most importantly, the framework enables direct feedback to the system design process by delivering model-based results to the design team instead of document-based reports. This facilitates more efficient iterative design and reduces the overall cost of system improvement.

3. PROPOSED FRAMEWORK

RAAML provides standardized constructs for modelling system risk, including items, dependencies, failure modes, and fault logic structures.

Based on the principles described above, the MBSE-driven framework establishes systematic relationship between system model and risk model. The relationships are defined not only at the meta-model level but also at the data level, ensuring that the system model serves as a single authoritative source of truth for subsequent RAM analysis.

3.1. Concept Mapping

The table below shows the concept mapping from SysML elements to RAAML elements.

Table 1. Typical Concept Mapping Table

SysML Element	RAAML Element	Meaning
Block	Item	Physical component
Part Property	Item Instance	Installed component
Connector	Failure Propagation Path	Fault propagation and critical path
Activity	Maintenance Task	Repair / maintenance logic
Internal Block Diagram	Architecture Model	System configuration
State Machine	Failure State Model	Failure mode
Requirement	Reliability Requirement	MTBF/RAM targets

Through the mapping rules, the main elements of risk models can be derived from system model. With necessary extra information input, including MTBF for each component, etc, the risk model can cover the whole system.

The transform framework should be described in different levels for engineering implementation, semantic level and meta-model level.

3.2. Semantic Model Mapping

The RAM analysis can be executed using derived risk model as below:

- Structural model→RBD
- Behavioural model→Mission Profile
- Interface + Structure→Failure Propagation Table

Based on the characteristics of SysML models, the mapping rules are categorised into different types, as summarised in below table.

Table 2. Semantic Model Mapping Table

Category	SysML Element	RAM Output
Structural Model	Part	Failure propagation Path
	Connection	RBD
Behavioural Model	Use case	Mission Profile
	Activity/Action	RBD
	State/Transition	RBD
	Sequence/Interaction	Mission Profile
Requirement Model	Requirement	RAM Target Risk constraints
Interface Model	Item Flow	Failure propagation Path
	Signal Flow	Critical Path
	Interface Network	Failure propagation Path

3.3. Meta-model Mapping

The meta-model mapping establishes explicit correspondences between basic elements of SysML and RAAML, including meta-classes, relationships, and constraints.

Rather than attempting to map the complete SysML v2 meta-model, the framework focuses on a RAM-relevant subset consisting of structural, behavioural, requirement, and interface/flow constructs, which are sufficient to generate the main RAM analysis, including RBD, mission profiles, and failure propagation models.

Figure 3 presents the ontology-level view of the transformation framework, which shows how a RAM-relevant subset of the SysML v2 meta-model is mapped to corresponding RAAML ontology elements through four categories of mapping rules, providing the basis for deriving RBDs, mission profiles, failure propagation graphs, and quantitative risk models.

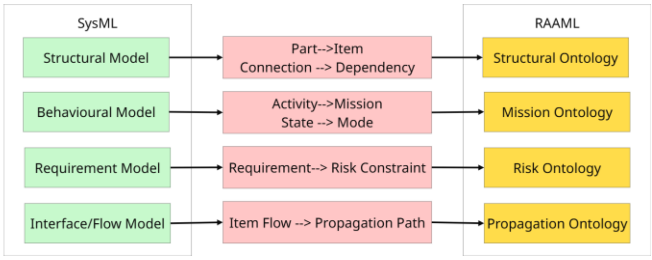


Figure 3. Ontology Mapping View

For clarity, the meta-model mapping is divided into several categories as below tables.

Table 3. Structural Mapping Table

SysML	**RAAML**	**RAM usage**
Part Definition	Item	Basic system element
Part Usage	Item Instance	Installed component
Connection Definition	Dependency	Structural dependency relation
Interface Definition	Interaction	Interface semantics
Structural hierarchy	Hierarchy	System decomposition
Multiplicity repeated parts	Redundancy structure	Parallel structure in RBD

Table 4. Behavioural Mapping Table

SysML	**RAAML**	**RAM usage**
Activity	Mission	Operational scenario
ActionDefinition ActionUsage	MissionPhase	Phase-based analysis
ControlFlow	Sequence	Execution order
State Definition	OperationalMode FailureMode	Mode Based RCM
Transition	StateTransition	Failure evolution

Table 5. Requirement Mapping Table

SysML	**RAAML**	**RAM**
Requirement Definition	Risk Requirement	RAM target
Requirement Usage	Requirement allocation	Traceability
Satisfy	Mitigation	Risk control
Verify	Validation Condition	Verification of RAM target
Refine	Requirement decomposition	Hierarchical constraints

Table 6. Interface and Flow Mapping Table

SysML	**RAAML**	**RAM**
Item Flow	PropagationPath	Failure transmission
Flow Connection	Propagation Link	Critical path
Port Definition	Interaction Point	Entry/exit of failure effects
Transfer	Propagation Channel	Energy / signal dependency
Interface network	Failure Propagation Graph	System-wide propagation

The approach enables a consistent and traceable RAM model to be derived directly from the system architecture. The following section presents a ROV case study to illustrate the application of the framework.

3.4. Transformation Rules

Based on the above different layers mapping table, the below are the transformation rules implement example:

```
For each PartDefinition p in SysML:
    create Item i in RAAML
For each Connection c between p1 and p2:
    create Dependency(p1 → p2)
For each ItemFlow f:
    create PropagationPath
```

4. ROV CASE STUDY

The case study considers the top-level deployment and operation chain required to provide power, control, communication, and lifting support for the ROV mission. The analysis boundary includes the following four main subsystems:

- ROV Control Equipment Rack
- Surface Power Supply Units (PSU)
- Main Lift, Tethers and Deck Leads (Umbilical)
- ROV

These four elements form the top-level architecture used for deriving the RAM risk model. Figure 4 shows the system architecture schematic diagram.

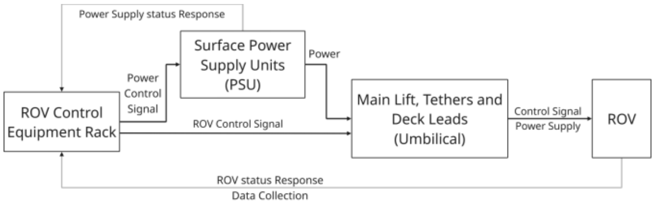


Figure 4. ROV System Architecture diagram

The framework does not rely on a one-to-one diagram conversion but instead derives RAM artefacts from the semantics of the system model. Structural elements provide the basis for reliability decomposition, interface and flow information define dependency and propagation relationships, and behavioural elements support mission-based reliability analysis. In this way, the generated RAAML model remains directly traceable to the original system architecture.

4.1. System Model

The top-level system consists of the ROV control equipment rack, surface power supply units, umbilical subsystem, and the ROV itself. Structural definitions provide the basis for system decomposition and reliability block diagram generation, interface and flow models capture electrical, communication, and mechanical dependencies, and behavioural models define mission phases from preparation to recovery. Reliability-related parameters, such as MTBF and MTTR, are introduced as external attributes to support quantitative analysis.

The SysML model serves as the source architecture for the automatic derivation of RAM risk models in the framework.

4.1.1. Structural Model

The structural model is the basis for deriving the top-level RBD.

The top-level part definitions are below:

```
part def ROVTopViewGroup;
part def ROVControlEquipmentRack;
part def SurfacePowerSupplyUnit;
part def Umbilical;
part def ROV;
```

The top-level composition is below:

```
part def ROVTopViewGroup {
   part controlRack : ROVControlEquipmentRack;
   part psu        : SurfacePowerSupplyUnit;
   part umbilical  : Umbilical;
   part rov        : ROV;
}
```

4.1.2. Interface and Flow Model

The interfaces and connections are the source for the failure propagation path with critical path identification, and sensor placement candidates:

The interface definitions are below:

```
interface def ElectricalPowerIF;
interface def ControlCommandIF;
interface def TelemetryIF;
interface def VideoDataIF;
interface def MechanicalIF;
```

4.1.3. Port Definitions

```
part def ROVControlEquipmentRack {
   port controlOut : ControlCommandIF;
   port telemetryIn : TelemetryIF;
   port videoIn : VideoDataIF;
}
part def SurfacePowerSupplyUnit {
   port powerIn : ElectricalPowerIF;
   port powerOut : ElectricalPowerIF;
}
part def Umbilical {
   port powerIn : ElectricalPowerIF;
   port powerOut : ElectricalPowerIF;
   port commandIn : ControlCommandIF;
   port commandOut : ControlCommandIF;
   port telemetryIn : TelemetryIF;
   port telemetryOut : TelemetryIF;
   port videoIn : VideoDataIF;
   port videoOut : VideoDataIF;
}
part def ROV {
   port powerIn : ElectricalPowerIF;
   port commandIn : ControlCommandIF;
   port telemetryOut : TelemetryIF;
   port videoOut : VideoDataIF;
   port MechanicalOut : MechanicalIF;
}
```

4.1.4. Connections

```
connect controlRack.controlOut to umbilical.commandIn;
connect umbilical.commandOut to rov.commandIn;
connect psu.powerOut to umbilical.powerIn;
connect umbilical.powerOut to rov.powerIn;
connect rov.telemetryOut to umbilical.telemetryIn;
connect umbilical.telemetryOut to controlRack.telemetryIn;
connect rov.videoOut to umbilical.videoIn;
connect umbilical.videoOut to controlRack.videoIn;
```

4.1.5. Behavioural Model

The activity structure is used to derive the mission phases with involvement duration to do mission-based reliability assessment.

Mission Use Cases are below:

use case PrepareROVSystem;
use case DeployROV;
use case ExploringROV;
use case RecoverROV;
use case PostDiveCheckROVSystem;

Activity Flows are below:

action InitializeControlRack;
action EnergizePSU;
action VerifyUmbilicalContinuity;
action LaunchROV;
action DescendToWorksite;
action PerformSubseaOperation;
action AscendFromWorksite;
action RecoverROV;
action PostDiveCheckSystem;

Mission Sequence are below:

PrepareROVSystem:
  InitializeControlRack
  -> EnergizePSU
  -> VerifyUmbilicalContinuity
DeployROV:
  LaunchROV
  -> DescendToWorksite
OperateROV:
  PerformSubseaOperation
RecoverROV:
  AscendFromWorksite
  -> RecoverROV
ShutDownROVSystem:
  ShutDownSystem

4.1.6. State Model

The state model supports the operational mode mapping and failure propagation path.
The top-level operational states are below:
state Idle;
state Powered;
state Deploying;
state Operating;
state Recovering;
state Faulted;
state PostDiveCheck;

The state transitions are below:

Idle -> Powered
Powered -> Deploying
Deploying -> Operating
Operating -> Recovering
Recovering -> Powered
AnyState -> Faulted
Faulted -> PostDiveCheck

4.1.7. Requirements Model

These requirements are transformed into RAM targets, risk constraints, validation conditions and the traceability link in the RAAML model.:

The following are reliability and safety requirements:

R1:The ROV system shall maintain continuous topside-to-subsea power delivery during mission operation.
R2:The ROV system shall maintain command and telemetry continuity throughout deployment, operation, and recovery.
R3:The system shall support safe recovery of the ROV after a detectable subsystem fault.
R4:The top-level deployment chain shall satisfy the required mission reliability target.
R5:The system shall support condition monitoring of mission-critical interfaces.

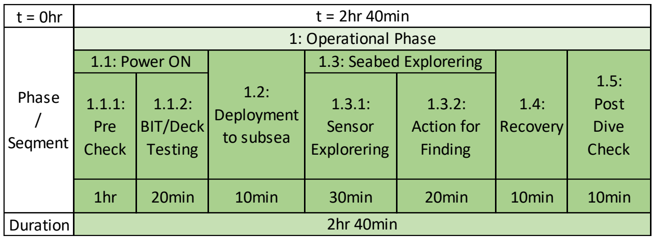
The requirements allocation examples are below:

R1 satisfied by psu, umbilical, rov.powerDistributionUnit
R2 satisfied by controlRack, umbilical, rov.onboardControlUnit
R3 satisfied by controlRack, umbilical, rov
R5 satisfied by controlRack, psu, umbilical

4.2. Risk Model

Through the transformation of framework, the system model elements are parsed and transferred to risk model accordingly. RAM analysis involves multiple interrelated elements within the risk model. Some representative models are presented as examples.

Figure 5 shows the mission profile according to the time-sequence diagram of ROV, which indicates each component's involvement with duration time in each mission cycle.



t = 0hr	t = 2hr 40min						
Phase / Seqment	1: Operational Phase						
	1.1: Power ON		1.2: Deployment to subsea	1.3: Seabed Explorering		1.4: Recovery	1.5: Post Dive Check
	1.1.1: Pre Check	1.1.2: BIT/Deck Testing		1.3.1: Sensor Explorering	1.3.2: Action for Finding		
	1hr	20min	10min	30min	20min	10min	10min
Duration	2hr 40min						

Figure 5. Mission Profile of ROV

Figure 6 illustrates the functional model with energy and signal flows, which is used to analyze failure propagation paths and identify critical dependencies within the system.

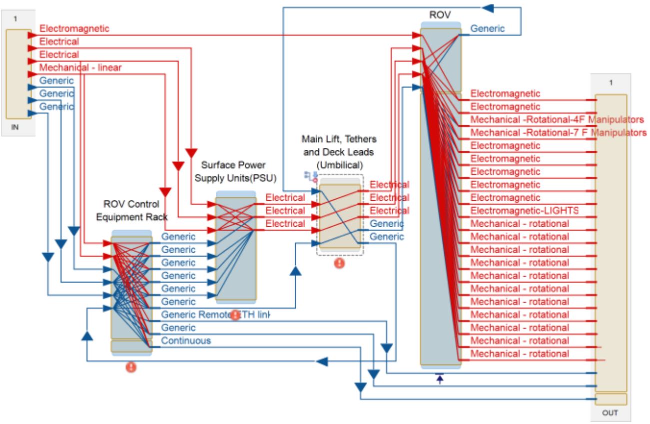


Figure 6. Functional Model for RAM Failure propagation

The model was created using MADe software, which adopts Fuzzy Cognitive Mapping (FCM) approach to represent system behaviour through causal relationships among material, energy, and signal flows. In contrast, SysML is primarily oriented towards describing system structure, interfaces, and nominal behaviour. MADe project emphasises functional dependencies and the propagation of failures across interconnected flows, thereby supporting the analysis of system degradation and failure effects in RAM and PHM contexts.

Figure 7 shows the top-level RBD of the ROV, together with the corresponding reliability evaluation results, providing a quantitative view of system-level reliability based on the derived model.

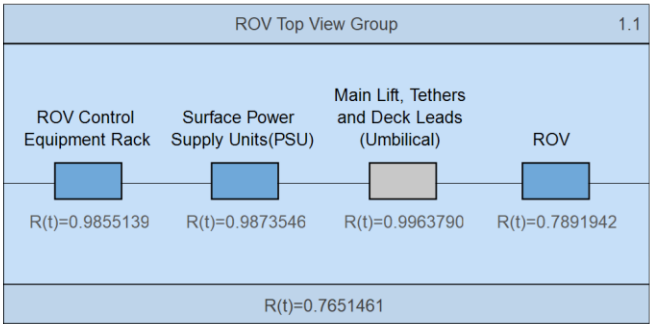


Figure 7. RBD Analysis for Reliability

Together, these results demonstrate that the framework can generate both structural and functional risk models directly from the system architecture, supporting consistent and traceable RAM analysis.

4.3. Effort analysis

Effort is measured in engineering man-hours required to construct functional and risk models. The baseline represents manual modelling based on design documentation, while the approach derives models directly from SysML architecture.

The below two figures show a consistent reduction in modelling effort when applying the SysML-to-RAAML transformation framework. The comparison between function modelling cost and RAM analysis cost indicates that the traditional approach requires a significant amount of manual effort to reconstruct functional and dependency models from design documentation. In contrast, the framework enables direct reuse of system models, thereby reducing redundant modelling activities.

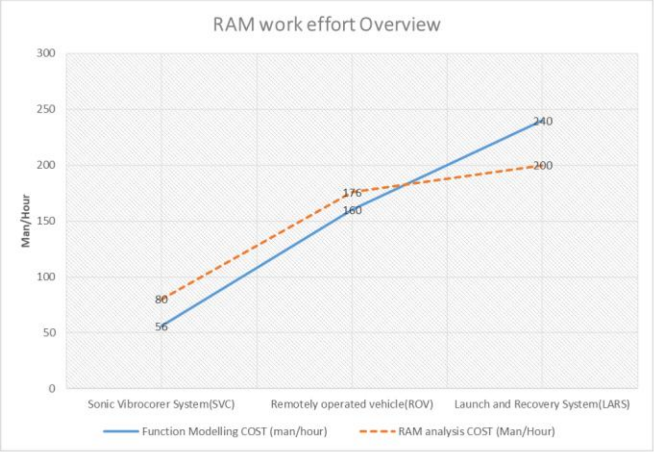


Figure 8. RAM work effort Overview

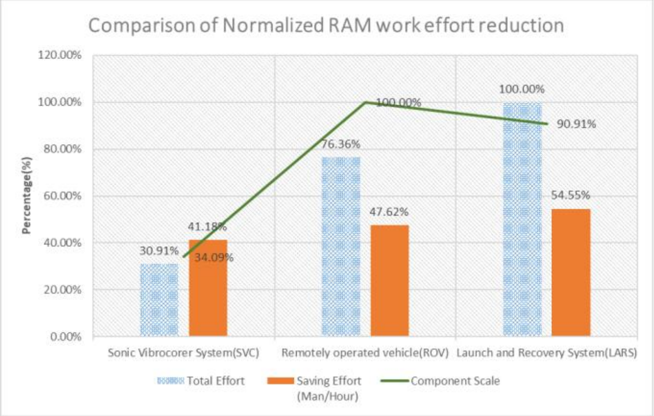


Figure 9. RAM work effort reduction comparison

For the SVC, the effort reduction is moderate, at approximately 34.09%, reflecting its relatively low system complexity and limited number of components. In this case, the benefit of the transformation is less pronounced compared to manual modelling.

For the ROV, the effort reduction increases to 47.62%. The higher system complexity, including multiple subsystems and interface dependencies, leads to greater modelling effort in traditional approaches, while the framework enables effective reuse of structured system information.

The most significant improvement is observed in the LARS, with a reduction of 54.55%. Although its structural complexity is not the highest, the lack of complete design documentation makes conventional risk modelling highly dependent on manual input and expert knowledge. The framework reduces this dependency by deriving risk models directly from system architecture, resulting in substantial effort savings.

Overall, the case study shows that the framework reduces RAM modelling effort by 34% to 55% across the evaluated systems. The benefit increases with system complexity and data incompleteness, as the approach eliminates the need to reconstruct functional and dependency models from documentation. By deriving risk models directly from SysML architecture, the framework improves modelling efficiency and scalability while maintaining consistency between system design and RAM analysis.

5. CONCLUSION AND FUTURE WORK

The paper presents a structured transformation framework for generating RAM risk models from SysML-based system architectures. The approach establishes a formal link between system design and risk analysis through concept, semantic, and meta-model mappings. By deriving reliability structures, mission profiles, and failure propagation relationships directly from system models, the framework improves traceability and reduces modelling effort.

The derived propagation paths and critical dependencies also provide a basis for sensor placement and condition monitoring design, supporting PHM applications.

The framework currently relies on external data for failure rates and maintenance parameters and does not explicitly model degradation mechanisms within SysML. Moreover, the transformation process does not yet cover all elements required for RAM analysis and still requires substantial manual effort from RAM engineers.

The case study shows that the approach reduces RAM modelling effort by 34% to 55%, with increasing benefits for more complex systems. The results confirms that the framework scales effectively and is particularly suitable for large maritime systems.

Future work will focus on integrating reliability and operational data sources and extending the framework to support dynamic risk assessment and PHM applications. In addition, benchmarking studies and additional evaluation metrics will be carried out to assess the completeness of the proposed approach. Building on these improvements, the impact on target-system performance and the demonstrated quality and safety contributions will be consolidated to further refine the framework.

ACKNOWLEDGEMENT

We would like to acknowledge Mr. Simon Crompton, head of Quality Engineering at Ocean Infinity, supporting the writing of the paper.

REFERENCES

OMG Systems Modeling Language™ (SysML®) Version 2.0 (2025), OMG Document Number: formal/2025-09-03, https://www.omg.org/spec/SysML/2.0/Language/

Biggs, Geoffrey & Juknevicius, Tomas & Armonas, Andrius & Post, Kyle. (2018). Integrating Safety and Reliability Analysis into MBSE: overview of the new proposed OMG standard. INCOSE International Symposium. 28. 1322-1336. https://doi.org/10.1002/j.2334-5837.2018.00551.x.

Diatte, K.; O'Halloran, B.; Van Bossuyt, D.L.(2022) The Integration of Reliability, Availability, and Maintainability into Model-Based Systems Engineering. Systems 2022, 10, 101. https://doi.org/10.3390/systems10040101

V. Sideris, Z. P. Oikonomou, S. Gerené, and A. A. Kana, (2025) "Advancing Model-Based Systems Engineering in the Development of Naval Vessel Systems Architecture." Systems Engineering: e70020. https://doi.org/10.1002/sys.70020

Risk Analysis and Assessment Modeling Language (RAAML) Libraries and Profiles Version 1.1 Beta 2, (2024) OMG Document Number: ptc/24-03-02, http://www.omg.org/spec/RAAML

RAAML Examples Risk Analysis Assessment Modeling (2021) https://www.omg.org/cgi-bin/doc?ptc/21-11-22.pdf

Hecht, M., Juknevičius, T., Post, K. and Armonas, A. (2025) "A Standard for Reliability Block Diagrams in SysML," 2025 Annual Reliability and Maintainability Symposium (RAMS), Destin, FL, USA, 2025, pp. 1-8, doi: 10.1109/RAMS48127.2025.10935201.

Hecht, M., Juknevičius, T., Post, K. and Armonas, A. (2024) OMG Standard to Extend SysML to Reliability Engineering. INCOSE International Symposium, 34: 2431-2451. https://doi.org/10.1002/iis2.13279

B. Kaiser, M. Soden and N. Heuermann, (2024) "A UAV Case Study on an MBSE Workflow with Integrated Modular Safety and Reliability Analysis," 2024 Annual Reliability and Maintainability Symposium (RAMS), Albuquerque, NM, USA, 2024, pp. 1-7, doi: 10.1109/RAMS51492.2024.10457654.

E. Brusa, (2021) "Digital Twin: Toward the Integration Between System Design and RAMS Assessment Through the Model-Based Systems Engineering," in IEEE Systems Journal, vol. 15, no. 3, pp. 3549-3560, Sept. 2021, doi: 10.1109/JSYST.2020.3010379.

Girard, Gaëlle & Baeriswyl, Ivan & Hendriks, Jonathan & Scherwey, Roland & Müller, Christian & Hönig, Philipp & Lunde, Rüdiger. (2020). Model based Safety Analysis using SysML with Automatic Generation of FTA and FMEA Artifacts. 5051-5058. 10.3850/978-981-14-8593-0_4941-cd.

W. Huang and D. Borja, (2025) "A Comprehensive System Level Reliability Assessment: Modeling Guided by FMEA," 2025 Annual Reliability and Maintainability Symposium (RAMS), Destin, FL, USA, 2025, pp. 1-6, doi: 10.1109/RAMS48127.2025.10935059.

M. W. Daniels and K. Pierre, (2023) "Integrating Reliability Engineering with Model Based Systems Engineering," 2023 Annual Reliability and Maintainability Symposium (RAMS), Orlando, FL, USA, 2023, pp. 1-6, doi: 10.1109/RAMS51473.2023.10088275.

Cvijic S., Harrison S., Marotta S. (2026) "Bridging The Gap Between MBSE And Reliability Analysis," Proceedings of the 72nd Annual Reliability & Maintainability Symposium (RAMS®), Miramar Beach, FL, 2026

Cvijic S., Hookway S., Harrison S.and Strelzoff A., (2025) "Probabilistic AI Reliability Analysis in MBSE and SysML," 2025 Annual Reliability and Maintainability Symposium (RAMS), Destin, FL, USA, 2025, pp. 1-6, doi: 10.1109/RAMS48127.2025.10935158.

Suls J., (2024) "Leveraging MBSE to Analyze Reliability, Availability & Maintainability for System-of-Systems," 2024 Annual Reliability and Maintainability Symposium (RAMS), Albuquerque, NM, USA, 2024, pp. 1-6, doi: 10.1109/RAMS51492.2024.10457636

Niculita, Ioan-Octavian; Jennions, Ian K.; Medina Valdez, Miguel. (2014) Towards an integrated COTS toolset for IVHM design. The 2nd European Conference of the Prognostics and Health Management Society, Nantes, France.11 p.

BIOGRAPHIES

Yanlai Wang is RAM engineer of Ocean Infinity, responsible for risk modelling for marine vessels and RAM analysis, who has over 15 years at MBSE system architecture development in aircraft industry. He has a MSc in Computer Science from Beijing University of Technology and a BSc of Xi'An Jiaotong University.

Toby Russell is head RAM engineer of Ocean Infinity, who has 16 years at sea with the majority as chief ETO/Chief Electrical Engineer but also serving as 2nd engineer. He holds a MSc in applied Instrumentation and Control, and he has a keen interest and passion for marine engineering that has progressed into the fields of RAM engineering to support the development of novel technologies to enable remote vessel operation on a commercial level. Toby has worked for Ocean Infinity for 4 years leading novel system development and integration projects as well as leading on development of RAM philosophies and strategies.

Octavian Niculita is a Senior Lecturer in Instrumentation with Glasgow Caledonian University. He has a PhD in Industrial Engineering from the Technical University of Iasi, Romania carried out under the EDSVS framework. His current research interests include industrial digitalisation, predictive maintenance, PHM system design, integration of PHM and asset design for aerospace, maritime, and oil & gas (surface and subsea) applications. He has over 15 years of experience in design and development of prognostics and health management applications, having worked on applied aerospace projects funded by The Boeing Company and BAE Systems as a Research Fellow and Technical Lead on his previous appointment with the IVHM Centre at Cranfield University, UK. He is a member of the Prognostics and Health Management Society, InstMC and the IET.