2023 NDIA MICHIGAN CHAPTER
GROUND VEHICLE SYSTEMS ENGINEERING
and TECHNOLOGY SYMPOSIUM
Digital Engineering/Systems Engineering (DE/SE)
Technical Session
August 15-17, 2023 - Novi, Michigan

ITERATIVE CO-DESIGN OF ORGANISATIONAL PROCESSES AND TOOLCHAINS FOR MODEL-BASED RELIABILITY, AVAILABILITY MAINTAINABILITY & SAFETY INTEGRATION

Jake Langton[1], Sam Hilton[2]

[1]PHM Technology, Melbourne, Australia
[2] PHM Technology, Melbourne, Australia

ABSTRACT

A toolchain must be functionally cohesive with a business process, especially in technical domains such as complex systems engineering. Despite the industry-wide shift towards model-based digitization within engineering organizations, there is a lack of development in implementing model-based RAMS (Reliability, Availability, Maintenance, Safety) processes. This results in a missed opportunity to create value throughout the entire system lifecycle, from conceptual design to operations. This paper proposes some reasons for this and outlines a framework for evaluating model-based toolchains in the context of the entire Engineering cycle. A model-based architecture for RAMS is proposed and contrastively evaluated with respect to SysML. Key use cases are identified, and benefits are demonstrated using Maintenance Aware Design Environment Software.

Citation: J. Langton, S. Hilton, "Iterative Co-Design Of Organizational Processes and Toolchains For Model-Based Reliability, Availability Maintainability and Safety Integration," In *Proceedings of the Ground Vehicle Systems Engineering and Technology Symposium* (GVSETS), NDIA, Novi, MI, Aug. 16-18, 2022.

1. INTRODUCTION

The iterative co-design of organizational processes and toolchains is crucial for managing complex organizations. Functionally cohesive toolchains enable the creation, exchange, aggregation, and reuse of structured expert domain knowledge, supporting business processes and reducing waste. The core driver of productivity and innovation is information exchange between stakeholders over time.

Model-based engineering aims to unify organizational silos and streamline deliverables and requirements across engineering design functions, facilitating iterative optimization of asset management during operations. Integrating Reliability, Availability, Maintainability, and Safety (RAMS) processes into the toolchain can enhance communication, collaboration, and decision-making, leading to better risk management and system performance.

In this paper, we address the importance of cohesive toolchains in complex systems engineering, the current state of model-based digitization, and the need for model-based RAMS process integration. We also provide a methodology for evaluating model-based engineering implementations and outline the key attributes of an ideal process-driven toolchain. A high-level semantic data RAMS architecture is proposed and assessed in the context of Model-Based Engineering (MBE) approaches, using a case study with the model-based RAMS tool MADe.

2. The Proposed Framework for Evaluating Model-Based RAMS Toolchains

2.1. Toolchain evaluation In Defense

The Department of Defense Architecture Framework (DODAF) [1] plays a critical role in evaluating model-based toolchains in the context of diverse stakeholder perspectives. This paper revises and extends the definitions of DODAF viewpoints, emphasizing the importance of communication between enterprise software users and developers, and the need for flexibility, integration, and risk management in model-based engineering platforms. The integrated toolset viewpoint is introduced to address challenges in the integration of modern and legacy toolsets, and the trade-offs between flexibility and risk of modelling error across different domains.

The definitions of these DODAF viewpoints and a new integrated toolset viewpoint have been provided below:

- **Capability Viewpoint:** The enterprise goals related to the entire vision for executing a specified course of action.
- **Operational Viewpoint:** The organizations, tasks, and information that must be communicated between stakeholders to accomplish a goal.
- **Services Viewpoint:** The system, service, and interconnection functionality that provides or supports operational activities.
- **Data Model Architecture Viewpoint**: Conceptual, Logical and Physical data models
- **Integrated Toolset Viewpoint (New):** The distributed collection of tools that provide services based on information stored in the data model.
- **Standards Viewpoint:** The set of rules governing the flow, structure and semantics of information as well as the arrangement, interaction, and interdependence of system parts.

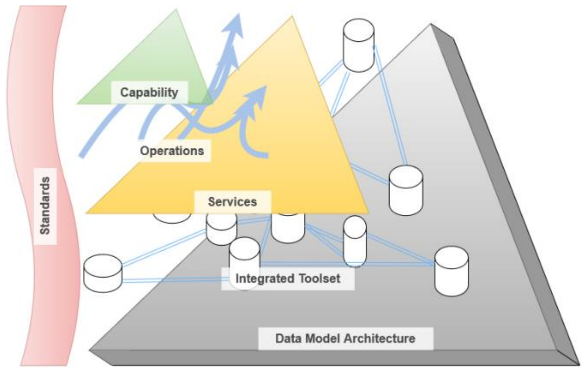

Figure 1: Variation Of DODAF Pyramid

Figure 1 presents a variation of the Department of Defense Architecture Framework (DODAF) tailored to evaluate model-based toolchains in the context of diverse stakeholder perspectives. The pyramid highlights the rapidly changing capability requirements of an organization, such as the need to transition to a holistic model-based approach, which is supported by a broader but less adaptive infrastructure base. As market conditions evolve, the demand for new capabilities exerts pressure on the processes, supporting services, and data architecture that empowers an organization to adapt its capabilities in response to these demands.

Given the impracticality of retroactively developing bespoke application services for any organization, the layered framework necessitates the introduction of a new viewpoint situated between the data architecture and services layers. The integrated toolset viewpoint acknowledges that the integration of various modern and legacy toolsets may create bottlenecks in the development of streamlined organizational processes, emphasizing the importance of addressing these integration challenges for enhanced operational efficiency. The varying requirements across different engineering domains necessitate the consideration of a variety of methods and languages that can be used to support MBE [2].

2.2. A Services Approach

Services create a layer that separates operational activities from organizational resource arrangements, such as personnel and information systems. Services create a layer that separates operational activities from organizational resource arrangements, such as personnel and information systems [1].

When available toolsets functionalities (business services) are unable to be decomposed into the atomic business process, either the processes become unnaturally constrained, or toolsets must be adapted [3]. Figure 2 depicts how business services provided by toolsets should be decomposed to complement the atomic business process of an organization. Each toolset A, B, and C represents an integrated service platform. The red dotted box linked to tool C illustrates how improperly decomposed business services result in additional work requirements (C.1.1) and vastly constrain the available integration opportunities with Tool B.

Given that the selection of tools influences an organization's processes and vice versa, it is essential to co-design integrated toolsets and business processes to optimize employee productivity and align it with clearly defined business objectives. To achieve this, organizations must establish multi-discipline modelling teams, assign responsibilities, complete modelling tasks, share common data, produce reliability artifacts, verify and refine models, and apply insights to future missions [4]. Similarly, Carroll and Malins [5] demonstrate that a mature, well-documented, and enterprise-wide systems engineering (SE) process is required, spanning from requirements development

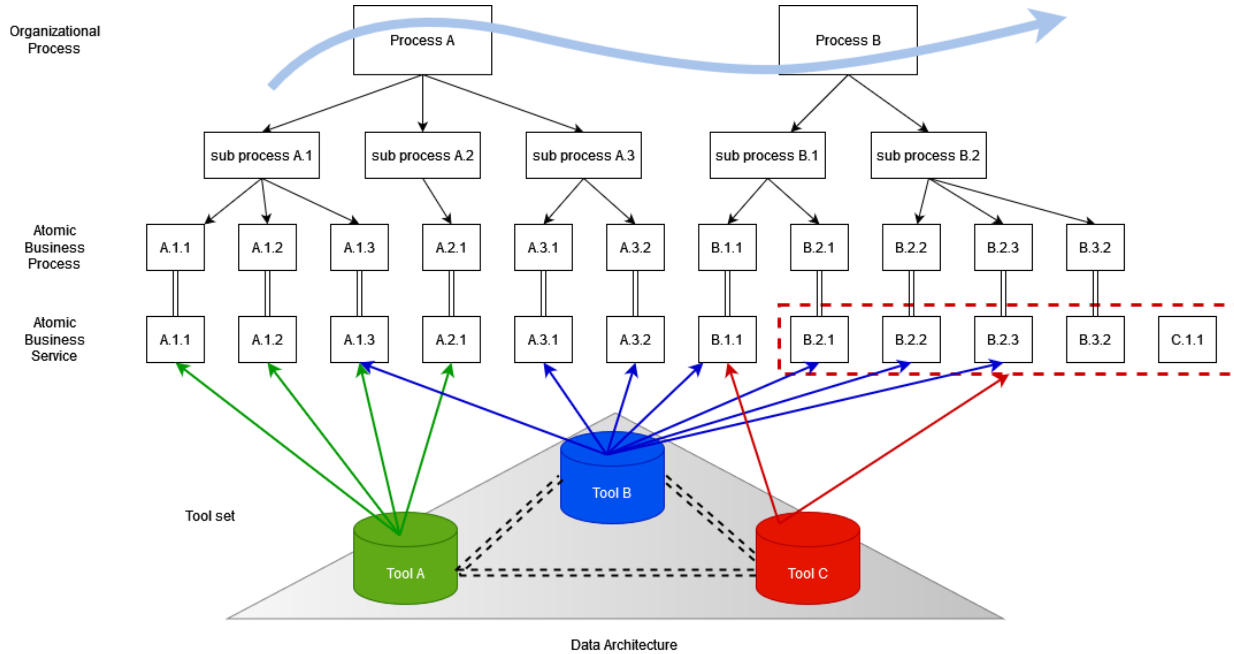

Figure 2: Toolset functionality constrains the ability to service organizational processes

and analysis through system testing, verification and validation (V&V). In addition, organizations must define MBSE model management processes to create, update, and maintain the MBSE models throughout the full lifecycle, derive engineering artifacts from the models at each stage of the system development lifecycle and institutionalize tool-use procedures to ensure compatibility across teams.

2.3. Additional Considerations

Functionally cohesive modules and seamless integration between services are crucial. While vertical integration can reduce customization costs, it may not be feasible for the entire project lifecycle. Toolsets should be designed for integration with other tools, requiring a common data architecture framework. Constant communication between enterprise software users and developers is vital for the iterative co-design of operational workflows and service capabilities, ensuring flexibility across projects, organizations, and lifecycles.

The inherent trade-off between flexibility and usability [6] is well known. In model-based engineering, it manifests itself by increasing the risk of modelling mistakes and therefore analysis errors. Inconsistencies are due to modelling language freedom or insufficient tool support risk of the spreading of faults and failures throughout development phases and among teams [7, 8] .

Carroll and Malins [5] identified the lack of skilled systems engineers, and skilled MBSE engineers, as a major hindrance to implementing an MBSE approach successfully – demonstrating the importance of toolset simplicity and usability. This is a sentiment echoed by Lindsey, et al. [4], who evaluated tools based on MBSE utilization, ease of use, and breadth of assurance discipline coverage. Diatte, et al. [9] proposed criteria of accessibility, ease of use, complex system and database integration.

2.4. RAMS Toolchain

Integrating RAMS with traditional MBSE approaches, such as SysML, is facilitated by applying an evaluation framework that focuses on an organization's architectural aspects, ensuring the selection of suitable tools that support necessary RAMS analysis capabilities. In addition to evaluating the ability RAMS of tools to produce project-specific services such as Failure Mode and Effect Criticality Analysis (FMECA), Fault Tree Analysis (FTA), Maintainability/Availability Analysis, and Probabilistic Risk Assessment (PRA) [4], the following general principles for toolchain evaluation must be assessed.

1. Flexible integration into varied organizational processes and standards
2. Ease of use and understanding
3. Seamless intra- and inter-organizational data-model integration.

3. MBSE, SysML and RAMS

3.1. SysML

SysML, or Systems Modeling Language, has become the de facto approach for model-based engineering due to its ability to capture complex system architectures and requirements. It provides the promise of standardized notation and semantics for creating models that can be shared and understood by all stakeholders. This common language aims to improve communication and collaboration between teams, reducing the risk of errors and inconsistencies in the system design through the use of automated analysis and verification tools, which can help identify design flaws earlier in the process and reduce the cost and time of system development [10].

SysML's flexibility and extensibility allow for customization but can also lead to inconsistencies and incompatibilities between different models, creating challenges in training, communication, and model integration. Capozucco [7] identified that language or tool-induced ambiguities can result in model errors and loss of benefits from a model-driven approach. Ambiguity and flexibility introduce challenges in training and communication among team members who may have different interpretations of the language's concepts and constructs. While SysML enables organizations to speak in one language it often results in significantly different dialects, resulting in a high cost of information transfer and reusability within an organization. SysML's flexible semantics often necessitate the creation of new models for each analysis, using various tools that reference SysML model elements [10]. A more tightly defined toolset can reduce training, communication, and model integration costs within specific organizational functions that don’t require the flexibility of systems engineering.

3.2. RAMS as a function of MBE

The ever-increasing complexity of engineering systems necessitates the expansion of Model-Based Engineering (MBE) beyond the realm of systems engineering. Many industry sectors, including subsea development, nuclear, satellite, and aviation, are grappling with inadequacies in their procedures for framing reliability, availability, maintainability, and safety (RAMS) within the design process, which is further exacerbated by the intricate nature of design solutions [11]. According to Dallosta and Simcik [12], total ownership costs (TOC) incurred during the operations and support (O&S) phase may constitute 65 percent to 80 percent of total life cycle cost (LCC), with design decisions establishing a "cost commitment" of approximately 70 percent of a system's LCC before any significant expenditures. Wolny, et al. [13]

identified the inability to align operational data with design models to support the execution and analysis of systems during runtime as a significant limitation of existing MBSE approaches. Implementing so-called “liquid models” [14], or Digital Availability Twins [15] can enable the closing of the loop between design and operational data collection, thus facilitating value creation across the entire lifecycle.

Model-Based Engineering (MBE) encompasses a wide range of engineering disciplines and utilizes various models and techniques not directly applicable to Systems Modeling Language (SysML). Examples of this include Function Flow Block Diagrams (FFBD) which are employed to support functional analysis or Markov chains which are similar to State machines but with notable distinctions in terms of the underlying mathematical framework [11]. Because analysis capabilities are directly influenced by the underlying data model implementing RAMS analysis in the context of MBE may involve several limitations. For instance, SysML/MagicDraw faces challenges in modelling hazard and operability analysis, FMECA, and reliability dependencies [11]. Lindsey, et al. [4] found that SysML/Magic Draw does not support RBD analysis, life analysis, critical items list (CIL), maintainability, or availability analysis. Additionally, no component per component and mission life probabilities were available for PRA and to derive adequate fault trees an additional set of state machines was required to be modelled, thus increasing work.

Zhang, et al. [11] highlight the need for an integrated framework between System Engineering models and RAM models that allow for qualitative and quantitative RAMS analysis to be iteratively completed throughout the design cycle (Figure 3). The elimination of inconsistencies between engineering teams was achieved through early identification of trade-offs and model-based trade studies, supported by findings that toolset integration with MBSE improved data modelling, information, analysis capabilities, and facilitated collaboration between RAMS and systems engineering [4, 9].

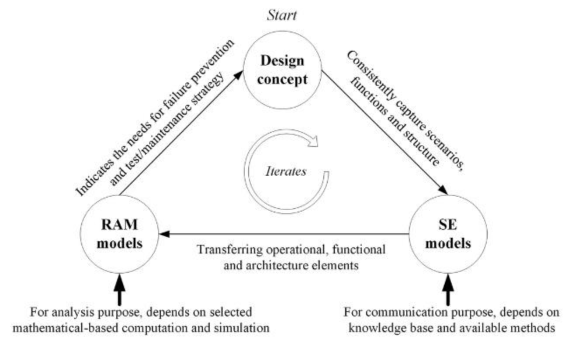

Figure 3 Integration of RAMS and Systems Engineering [11]

In addition to the lack of supporting capabilities, RAMS faces additional organizational and cultural headwinds that help explain why the digitization of RAMS has been slower compared to systems engineering. RAMS lacks the commercial incentives of model-based systems engineering which delivers significant cost and time savings and improves system reusability within future designs to deliver a return on investment.

Incentivizing the development of a model-based RAMS approach poses challenges, as operators primarily reap the benefits by utilizing designers' domain knowledge, such as system configuration and operational modes, to refine maintenance and operation practices based on operational data. As a result, the sector has not institutionalized digitization and model-based knowledge, making it more difficult and expensive to invest in processes and people capable of delivering model-based RAMS artifacts. This is a mistake.

In addition to streamlining the development of model-based RAMS deliverables [16], designers can capitalize on their extensive system knowledge by monetizing a RAMS baseline model that safeguards intellectual property and empowers operators to enhance operational efficiency through iterative analysis. To bridge the gap between operator priority requirements and designer capabilities the concept of CBM+ [17] has been identified as a priority use case to motivate the joint effort to develop exchangeable RAMS models.

4. Model-Based RAMS Architecture

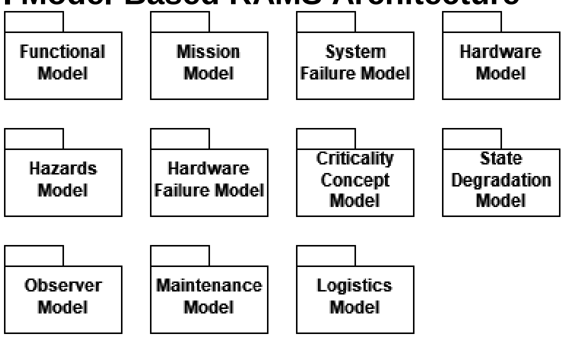

Figure 4: Proposed Functionally Cohesive model structure for RAMS.

This section proposes a functionally cohesive data architecture, where the separation of information reflects the functional separation of well-defined engineering disciplines across the lifecycle. The data architecture enables subject matter experts to digitize their domain knowledge in the form of well-structured interpretable models that interface with their related fields to enable the exchange of context and information for repeatable model-based analysis. The packages, each of which defines a distinct semantic data model can be modelled and exchanged throughout a model-based project toolchain that included SysML and physics simulation models.

The **Mission Model** captures the sequence of required functions and hardware over a mission, considering the effects of loading and environmental factors on hardware reliability. The **Functional Model** interfaces with SysML for function definition and identifies hazards and functional failures during conceptual development. The **System Failure Model** maps the flow of functional failure properties through an interconnected set of components and subsystems and enables functional redundancy conditions which mitigate the loss of function. The **Hardware Model** creates a comprehensive understanding of part and assembly hardware from CAD and associates these elements with attributes such as reliability, cost and logistics delay. The **State Degradation Model** models the effect of system function degradation on performance, cost, and fault observability.

Hazards Models capture concepts such as hazard sources, hazard mechanism, targets and mishaps as well as their dependencies due to common causes or cascading effects. Similarly, the **Hardware Failure Model** helps identify failure modes and their dependencies that manifest within the system, enabling hardware-based safety trade studies, root cause analysis and conduct Reliability Centered Maintenance (RCM) in the context of the entire system. **Criticality Concept Model** defines qualitative metrics such as severity, probability of failure and detectability and the control measures that mitigate events and conditions defined in hazards model and hardware failure models. The **Observer Model** mechanisms and constraints for observing measurable events or symptoms that manifest in response to a hazardous event of failure. The **Maintenance Model** captures various maintenance actions, such as scheduled, breakdown, and condition-based inspections while accounting for resource requirements and dependencies between actions. It is used to conduct trade studies and optimize maintenance allocation in the context of mission requirements and resource availability and logistics support modelled in the **Logistics Model.**

5. Case Study

5.1. Maintenance Aware Design Environment (MADe)

A Digital Risk Twin (DRT) for Reliability, Availability, Maintainability and Safety (RAMS) can be developed using MADe software by leveraging the aforementioned data architecture. The DRT provides a design risk model of the asset, enabling a repeatable and objective RAMS analysis through model-based system dependency and encoding multi-domain knowledge throughout the entire system life-cycle. MADe can produce Failure Modes, Effects, and Criticality Analysis (FMECA) [18], Diagnostic rules [19, 20], Reliability-Centered Maintenance (RCM) [21] as well as Fault Tree Analysis (FTA), Reliability Allocation, Reliability Block Diagrams, Markov Analysis, and Reliability/Availability Analysis [4].

MADe software combines a suite of interlinked models that each capture different RAMS domains, creating a DRT of the system. This facilitates a natural division of model development and analysis responsibility based on typical engineering roles while leveraging cross-domain commonality of information. The core capability of the DRT lies in its explicit modelling of system dependency simulation in the context of System Safety & Risk Assessment.

The DRT enables engineers to systematically approach "What If?" analysis, generating a generalized model of the expected system functional behaviour in the presence of failure. This design failure model informs various engineering decisions, including system configuration, redundancy arrangements, and mitigation measures to control and reduce risk. Compensating provisions and detection-based measures can also be applied to reduce risk impact.

5.2. Methodology

We propose a case study of an armored personal carrier (APC) to investigate the application of MADe software in the context of model-based RAMS data architecture. This case study will focus on the development of a system using MADe, demonstrating the required inputs and benefits of the following use case: Condition-Based Maintenance Plus (CBM+), which requires integration of Failure Modes, Effects, and Criticality Analysis (FMECA), Reliability-Centered Maintenance (RCM), and Diagnostics capabilities [17] . The methodology for this case study will involve creating a comprehensive model in MADe, defining system components, functions, and interdependencies. The model will then be used to perform FTA, FMECA, RCM, and Diagnostics analyses to identify potential failure modes, root causes, and maintenance requirements. The case study will also explore the benefits of using a standardized taxonomy for reporting failures, symptoms, and root causes of maintenance, enabling improved communication and data consistency across different teams and stakeholders. Furthermore, the case study will demonstrate the identification of functionally critical components based on mission requirements, allowing for more effective maintenance prioritization and informed decision-making throughout the system's life cycle. By showcasing the capabilities of MADe in the context of model-based RAMS data architecture, this case study aims to validate the software's potential for streamlining RAMS processes and enhancing overall system performance.

5.3. Model

The APC model was developed with the aid of functional schematics and requires a basic understanding of system operation and local modes of hardware failure and the concepts that lead to them. These models correspond to the system failure model (Figure 5) and the hardware failure model (Figure 6) of the proposed RAMS architecture respectively. Sensors can be assigned to flows of the system failure model to develop an observer model that is required for further diagnostic analysis and development of diagnostic rules. Subject matter experts assign criticality (Figure 7) to specific failure concepts in accordance with the procedures and standards of the project. These models are inherently linked with a hardware model that defines component reliability and can interface with mission models, functional models and maintenance models to facilitate workflows that are out of the scope of this paper.

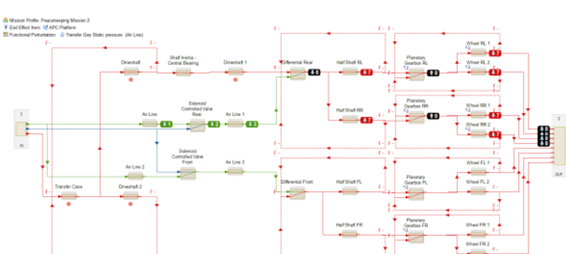

Figure 5: System Failure Model represented by a logical model within MADe.

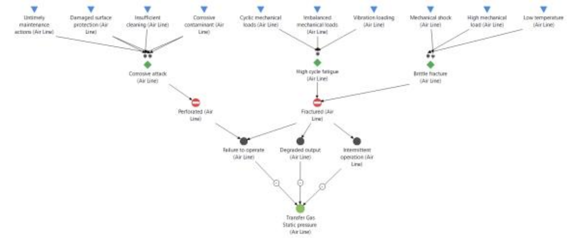

Figure 6: Hardware failure model represented in MADe by the failure diagram depicting the causes, mechanism, and faults of functional failure.

Figure 7: Criticality Model associates qualitative parameters with failure concepts defined in the MADe failure diagram.

The case study demonstrated how a simple model of connected functional components with failure diagrams and criticality values enabled the generation of integrated analysis capabilities of RCM, FMECA, FTA and a diagnostics analysis. Figure 8 illustrates how data reuse with MADe enables consistency and efficiency across these deliverables.

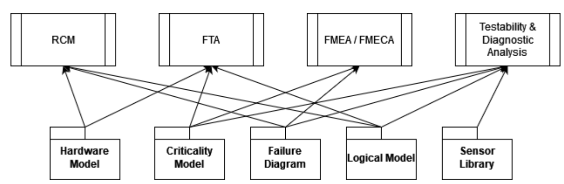

Figure 8: Figure Illustrates how data-reuse can be enabled across a range model-based analysis in MADe.

With respect to the toolchain criteria defined in 2.4, MADe demonstrates flexibility by supporting various standards such as MIL-STD-1629A, ISO226262, and SAE ARP 4761 for FMECA, and enabling the automated generation of model-based analyses that can flexibly integrate into variety of RAMS processes. The platform is easy to use, providing consistent model definitions across organizations, graphically distinct data elements, a taxonomy-guided model development process, and a consistent language for straightforward interpretation across RAMS domains. Seamless data integration is achieved through a web-based API that allows for importing, exporting, and syncing with external databases, as well as a SysML plugin for incorporating MADe functional definition stereotypes.

6. Conclusion

In conclusion, this paper presented a framework for evaluating process-based toolchains, focusing on the integration of RAMS with traditional MBSE approaches like SysML. The flexibility of SysML, while beneficial for various applications, was identified as a weakness when applied to RAMS-specific use cases. The main findings highlighted MADe's capability to support the model-based RAMS architecture defined in this paper, enabling data reuse and facilitating key CBM+ functionality within an intuitive modeling environment.

The implications of systems engineering across the lifecycle necessitate the integration of RAMS toolsets with SysML, emphasizing the importance of a comprehensive and robust approach to address the challenges associated with developing and managing complex systems. By leveraging the strengths of both SysML and specialized RAMS tools like MADe, organizations can enhance their ability to develop, analyze, and maintain reliable and safe systems throughout their lifecycles.

7. REFERENCES

[1] C. I. O. U.S. Department of Defense. "DODAF - DOD Architecture Framework Version 2.02 " https://dodcio.defense.gov/library/dod-architecture-framework/ (accessed 20/03/2023.

[2] P. Micouin, "MBSE, What is Wrong with SysML -First Issue," 2019.

[3] P. Knijnenburg, *Enterprise Application Integration Architecture*. Belgium: OSIRIS its, 2018.

[4] N. Lindsey, M. Alimardani, and L. Gallo, *Reliability Analysis of Complex NASA Systems with Model-Based Engineering*. 2020, pp. 1-8.

[5] E. R. Carroll and R. J. Malins, "Systematic Literature Review: How is Model-Based Systems Engineering Justified?," Office of Scientific and Technical Information (OSTI), 2016. [Online]. Available: https://dx.doi.org/10.2172/1561164

[6] W. H. Lidwell, Kritina and J. Butler, 2, Ed. *Universal principles of design*. Beverly, MA: Rockport, 2010.

[7] C. Capozucco, "CONSTRAINTS FOR AVOIDING SYSML MODEL INCONSISTENCIES," 2019.

[8] M. Alenazi, N. Niu, and J. Savolainen, "SysML Modeling Mistakes and Their Impacts on Requirements," 2019.

[9] K. Diatte, B. O'Halloran, and D. L. Van Bossuyt, "The Integration of Reliability, Availability, and Maintainability into Model-Based Systems Engineering," *Systems*, vol. 10, no. 4, p. 101, 2022, doi: 10.3390/systems10040101.

[10] S. Friedenthal, A. Moore, and R. Steiner, *A Practical Guide to SysMl*. 225 Wyman Street, Waltham, MA, 02451, USA: Elsevier, 2015.

[11] J. Zhang, C. Haskins, Y. Liu, and M. A. Lundteigen, "A systems engineering-based approach for

framing reliability, availability, and maintainability: A case study for subsea design," *Systems Engineering*, vol. 21, no. 6, pp. 576-592, 2018, doi: 10.1002/sys.21462.

[12] P. Dallosta and T. Simcik, "Driving Reliability, Availability, and Maintainability in While Driving Cost Out," *Defense AT&L, Special Issue*, 2012.

[13] S. Wolny, A. Mazak, C. Carpella, V. Geist, and M. Wimmer, "Thirteen years of SysML: a systematic mapping study," *Software and Systems Modeling*, vol. 19, no. 1, pp. 111-169, 2020, doi: 10.1007/s10270-019-00735-y.

[14] A. Mazak and M. Wimmer, "Towards Liquid Models: An Evolutionary Modeling Approach," in *2016 IEEE 18th Conference on Business Informatics (CBI)*, 29 Aug.-1 Sept. 2016 2016, vol. 01, pp. 104-112, doi: 10.1109/CBI.2016.20.

[15] S. Hilton, J. Langton, P. Conroy, and C. Stecki, "Digital Availability Twin – Targeted risk mitigation from design to operation " presented at the Reliability and Maintenance Symposium (RAMS) 2023, Orlando, Florida 2023.

[16] A. Thorn, P. Conroy, D. Chan, and C. Stecki, "The Digital Risk Twin – Enabling Model-based RAMS," presented at the Reliability and Maintainability Symposium (RAMS) 2023, Orlando, Florida 2022.

[17] *Reliability Centered Maintenance (RCM)*, D. o. Defense, 2011. [Online]. Available: https://www.wbdg.org/FFC/DOD/DODMAN/415122-M.pdf

[18] S. Rudov-Clark and J. Stecki, "The language of FMEA: on the effective use and reuse of FMEA data," *Australian International Aerospace Congress*, 01/20 2009.

[19] O. Niculita, O. Nwora, and Z. Skaf, "Towards Design of Prognostics and Health Management Solutions for Maritime Assets," *Procedia CIRP*, vol. 59, pp. 122-132, 2017, doi: 10.1016/j.procir.2016.10.128.

[20] S. Rudov-Clark, A. Ryan, C. Stecki, J. Stecki, and A. Hess, "Extending advanced failure effects analysis to support Prognostics and Health Management," 2010 2010: IEEE, doi: 10.1109/phm.2010.5413407. [Online]. Available: https://dx.doi.org/10.1109/phm.2010.5413407

[21] M. Cutajar and H.-B. Kim, "Streamlining Classical RCM using a Digitized Model-based approach " presented at the Realiability And Maintenance Symposium (RAMS) 2023, Orlando, Florida, 2023.