ID Specific workshop objectives at operational level
O-2 Limitations of current MBSE approaches and ways to circumvent or resolve these (e.g. through customization of processes and tools)

Update on the Applications of Model-Based Systems Engineering for JAXA's Engineering Test Satellite-9 Project

Yuta Nakajima[1*]*; Tsutomu Fukatsu*[2]

[1]*Japan Aerospace Exploration Agency, Tsukuba, Ibaraki, Japan, nakajima.yuta@jaxa.jp*
[2]*Japan Aerospace Exploration Agency, Tsukuba, Ibaraki, Japan*

1. Introduction

JAXA's Engineering Test Satellite-9(ETS-9) project team is applying Model-Based Systems Engineering (MBSE) to risk management of flight system development. ETS-9 demonstrates advanced technologies for the next generation communication satellite[1]. ETS-9 demonstrates the all-electric spacecraft technologies including the newly developed Hall Effect Thruster System as shown in Figure 1[2].

The Hall Effect Thruster System consists of three main components: the thruster, the power processing unit (PPU), and the propellant flow control module. The power processing unit controls and monitors thruster system performance. The comprehensive understanding of a complex Hall Effect Thruster system is a challenging issue for project systems engineers due to the complex interaction between components developed by different providers.

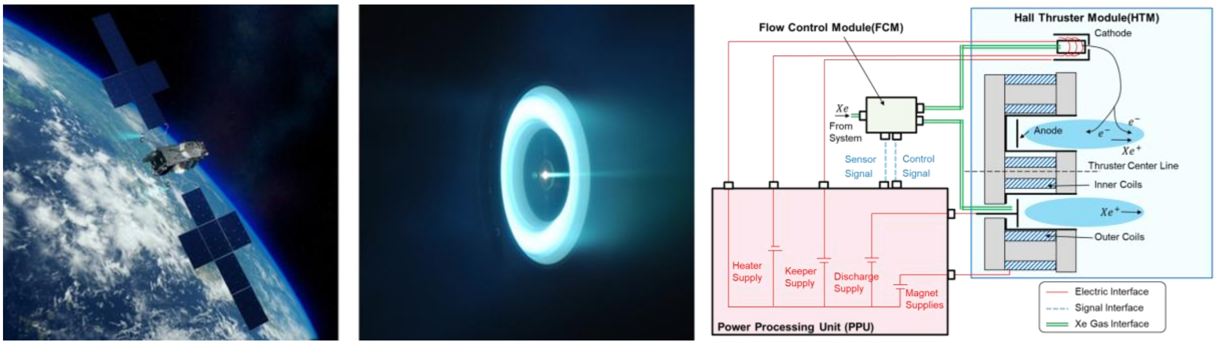


Figure 1. Artist concept for ETS-9 Mission (Left) and Hall Effect Thruster (Middle and Right)

2. Current limitations and solutions of MBSE for failure mode analysis

We leverage MBSE to manage the system complexity by using the formalized descriptions of system requirements, behaviors, and behavior allocations to system elements with SysML as shown in Figure 2[3]. We describe the system behaviors as sets of interactions between components by using the object flow of activity models. The object flow includes the electric energy, the xenon gas, information, and the force generated by the thruster. This behavior model is expanded to the failure mode analysis by customised descriptions of the failure modes.

Previous MBSE practice[3] has been focused on rigor description of system behaviors using a precise language. The model guides the system engineer to perform system-level failure mode analysis through query and visualization of the model. We extract the failure mode and analyse failure propagations by focusing on the chain of the interactions described in the system model. However, the previous approach lacks automation and reproducibility for function and failure propagation analysis.

We build an exploratory system architecture analysis framework that integrates the SysML-based MBSE process with R language-based data analytics as shown in Figure 3. We combine current MBSE practice and an exploratory data analysis approach[5] that is widely used in R language community. Rigor description of system architecture is transformed into graph-structured data that can be visualized and analysed in R language. This framework opens a new door for systems engineers to use an abundant resource in data science community.

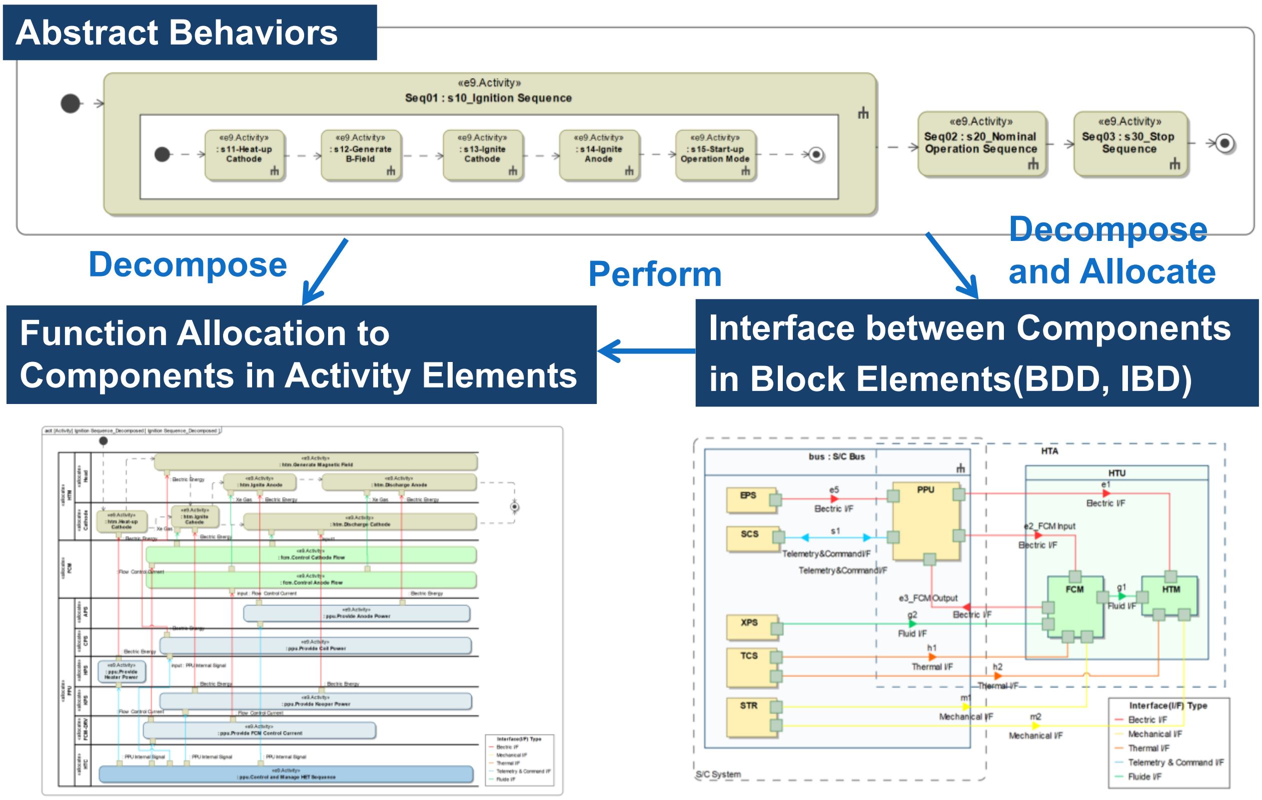


Figure 2. SysML architecture model for Hall Effect Thruster of ETS-9

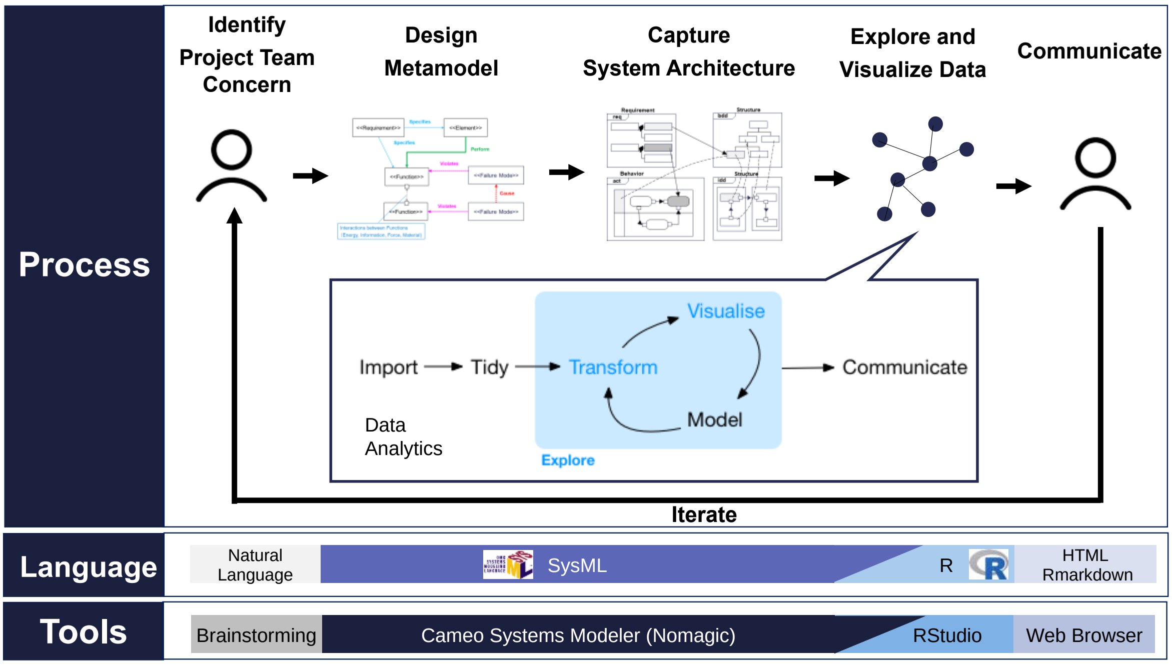


Figure 3: MBSE-driven Exploratory System Architecture Analysis Framework

3. Potential at scale

We build the solution for seamless integration of R and SysML languages. This solution enables systems engineers to leverage the rich analysis and visualization resources in data science community. We present two use cases.

Figure 4 shows the application of advanced data analytics to perform function propagation analysis. We treat SysML behavior and structure information as structured semantic data and transform these data into graph data in R language. We applied the Design Structure Matrix approach[4] to identify function propagations explicitly.

Figure 5 shows the example of process automation to generate FTA as a standard RAMS artifact. We transform SysML failure mode information as structured semantic data in R language. Then we transform and visualize data as FTA using open-source fault analysis packages[6].

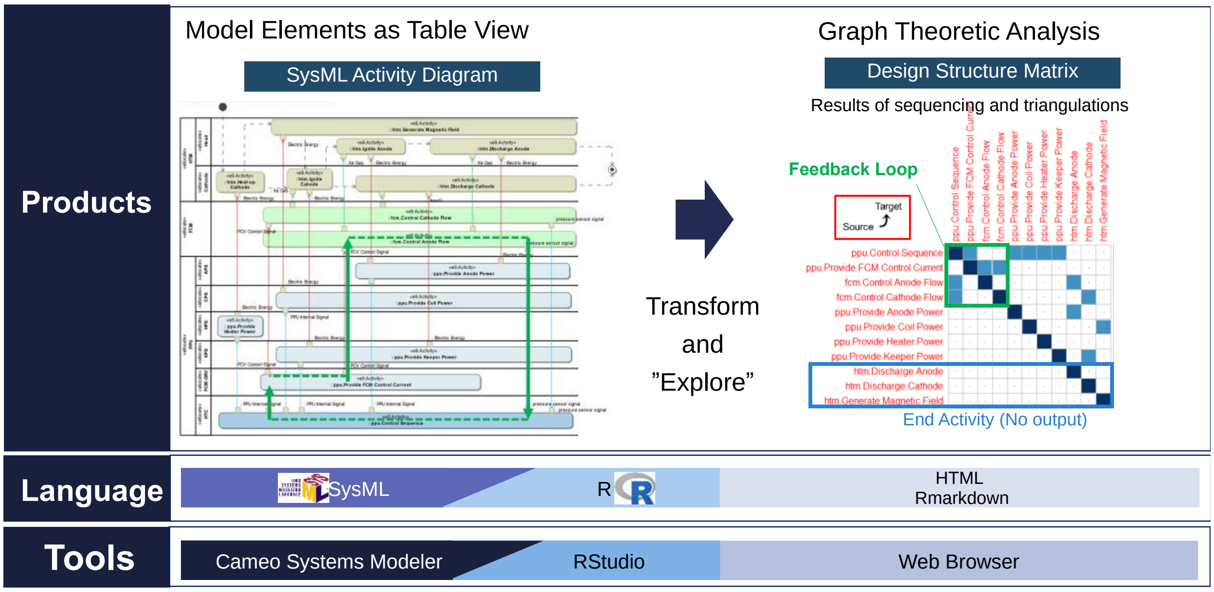
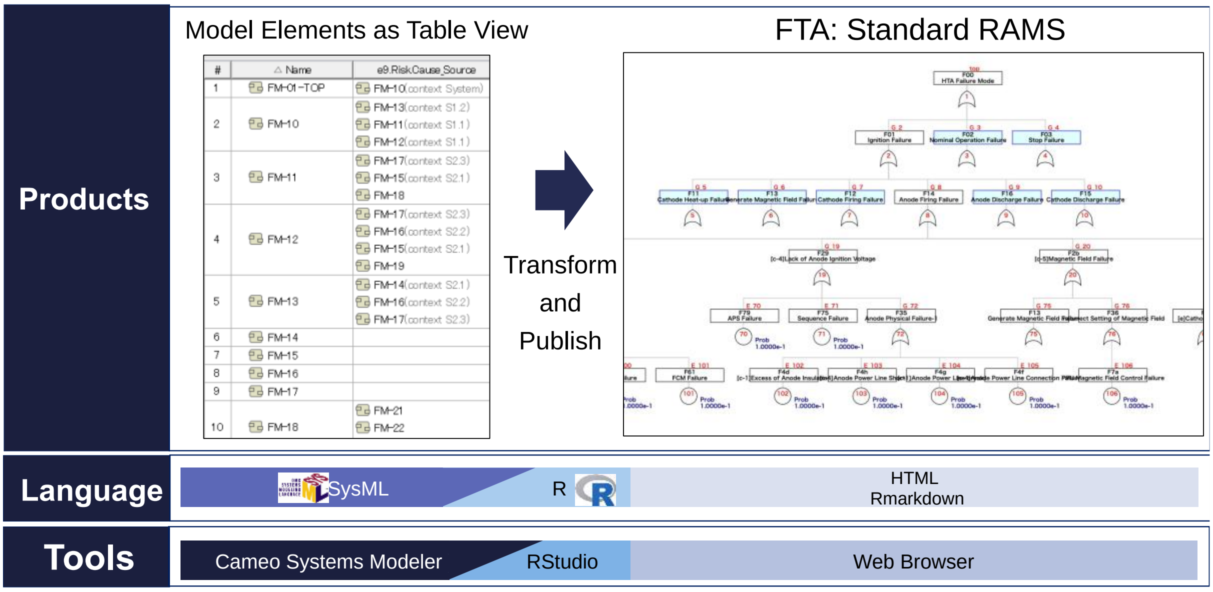


Figure 4: Application of advanced data analytics for system architecture analysis

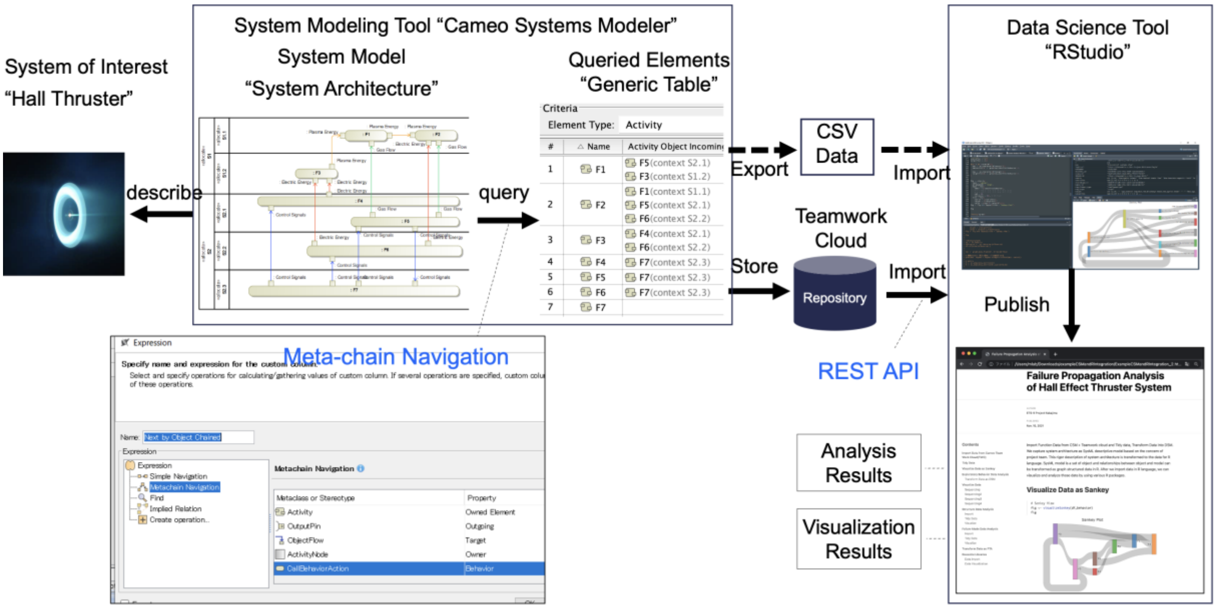


#	△ Name	e9.RiskCause_Source
1	FM-01-TOP	FM-10(context System)
2	FM-10	FM-13(context S1.2) FM-11(context S1.1) FM-12(context S1.1)
3	FM-11	FM-17(context S2.3) FM-15(context S2.1) FM-18
4	FM-12	FM-17(context S2.3) FM-16(context S2.2) FM-15(context S2.1) FM-19
5	FM-13	FM-14(context S2.1) FM-16(context S2.2) FM-17(context S2.3)
6	FM-14	
7	FM-15	
8	FM-16	
9	FM-17	
10	FM-18	FM-21 FM-22

Figure 5: Process automation to generate FTA as a standard RAMS artifact

4. Lessons Learned

We used No Magic's Cameo Systems Modeler as the SysML modeling tool and RStudio as the R language development environment. Since direct data exchange between these tools was not possible, data transfer between different tools has been realized manually via CSV data, which is a limitation of data analysis automation. We decided to exchange data via No Magic's Teamwork Cloud, which is a central repository for storing models as shown in Figure 6. We used meta-chain navigation to query chains of model elements. We developed an API to import SysML data into RStudio and reusable functions to automate the process of data tidying, analysis, and visualization. This allows for agile exploratory system architecture analysis.



Criteria
Element Type: Activity

#	Name	Activity Object Incoming
1	F1	F5(context S2.1) F3(context S1.2)
2	F2	F1(context S1.1) F5(context S2.1) F6(context S2.2)
3	F3	F4(context S2.1) F6(context S2.2)
4	F4	F7(context S2.3)
5	F5	F7(context S2.3)
6	F6	F7(context S2.3)
7	F7	

Expression

Specify name and expression for the custom column.
Select and specify operations for calculating/gathering values of custom column. If several operations are specified, custom colu of these operations.

Name: Next by Object Chained

Expression
- Expression
  - Simple Navigation
  - Metachain Navigation
  - Find
  - Implied Relation
  - Create operation...

Metachain Navigation

Metaclass or Stereotype	Property
Activity	Owned Element
OutputPin	Outgoing
ObjectFlow	Target
ActivityNode	Owner
CallBehaviorAction	Behavior

Figure 6: Integration of System Modeling Tool and RStudio

5. Conclusion

We present the effective use of MBSE application to the failure mode analysis in the implementation phase of the ETS-9 flight project. We find that the proposed data-driven risk analysis approach is reproducible and comprehensive compared with the traditional document-based approach that depends on the expert's knowledge and experience.

6. References

[1] Fukatsu, T., Ogawa, R., Sano, T., and Odawara, Y., "Development status of Engineering Test Satellite-9," Proceedings of 65th Space Sciences and Technology Conference, The Japan Society for Aeronautical and Space Sciences, 3D03, 2021 (in Japanese).

[2] Funaki, I., Sano, T., and Fukatsu, T., "Development Status of 6-kW-class Hall Thrusters at JAXA," 36th International Electric Propulsion Conference, Vienna, Austria, IEPC-2019-441, 2019.

[3] Nakajima, Y., and Fukatsu, T., "Applications of Model-Based Systems Engineering for JAXA's Engineering Test Satellite-9 Project," Model-Based Systems and Software Engineering Workshop 2020, European Space Agency, 2020.

[4] Browning, Tyson R. "Applying the design structure matrix to system decomposition and integration problems: a review and new directions." IEEE Transactions on Engineering management 48.3 (2001): 292-306.

[5] Wickham, H., and Garrett, G., "R for data science: import, tidy, transform, visualize, and model data," O'Reilly Media, Inc.", 2016.

[6] Package 'FaultTree': https://cran.r-project.org/web/packages/FaultTree/FaultTree.pdf [retrieved 12 September 2022]