2024 NDIA MICHIGAN CHAPTER
GROUND VEHICLE SYSTEMS ENGINEERING
AND TECHNOLOGY SYMPOSIUM
DIGITAL ENGINEERING / SYSTEMS ENGINEERING TECHNICAL SESSION
AUGUST 13-15, 2024 - NOVI, MICHIGAN

How Did General Veers Get His AT-AT? Product Line Engineering (PLE), Of Course![MH1]

Matthew Gagliardi[1], Matthew Hause[1]

[1]System Strategy Inc., Detroit, MI

ABSTRACT

MBSE and architecture frameworks were created to help systems engineers cope with the complexity of systems and systems of systems (SoS). Using these modeling languages, systems engineers can define the system, its context, connecting systems, internal components and behaviors, constraints, how the system changes over time, etc. However, when dealing with a family of systems or a product line of systems, one needs to understand the system variability, common components and behavior across the product line. This can decrease costs for development, maintenance, supply chains, and shorten development lifecycles. Product Line Engineering (PLE) is the engineering and management of a group of related products using a shared set of assets and a means of design and manufacturing. PLE can include system and software assets and involves all aspects of engineering including electrical, electronic, mechanical, chemical, etc. Model-based Product Line Engineering combines the best of MBSE and PLE. Using an example system from Star Wars, this paper will describe MB-PLE, the process for creating product lines, and the benefits of this approach as applicable to the military ground vehicle domain. Finally, it will show how the adoption of MB-PLE provides analysis of alternatives for candidate architectures.

Citation: Matthew Gagliardi, Matthew Hause, "General Veers Triumphant! - Configurable Architectures with Product Line Engineering (PLE)" In *Proceedings of the Ground Vehicle Systems Engineering and Technology Symposium* (GVSETS), NDIA, Novi, MI, Aug. 13-15, 2024.

1. INTRODUCTION

At the 2023 GVSETS Symposium, the authors presented "Implementing Mission Engineering with UAF" to demonstrate how a fictitious battle, and its corresponding people and systems, could be used to demonstrate mission engineering. [1] The model showed how a complex system of systems could be deployed in a manner similar to the Joint Forces operations employed by multiple militaries through the world. This scenario involved systems familiar to even the most casual of Star Wars fans. One of these systems was the All Terrain-Armored Transport or the AT-AT. Those more familiar with the films would have seen AT-ATs in multiple environments in addition to Hoth such as the forest planet of Endor and the arid planet of Krait. Like variants of existing combat vehicles, the AT-ATs have different configurations that enable them to operate in different environments and provide different capabilities for different missions. And as in real combat operations, the Empire learns operational lessons that they then apply as design changes to their systems. In the Battle of Hoth for example, the SnowSpeeders used towlines to encircle the legs of the AT-ATs causing them to fall over making them far easier to destroy. The Empire quickly added cutting blades to the leg armor in certain variants to render this attack strategy ineffective. [2]

1.1. The Problems with Multiple Models

One way to represent the different configurations of systems is to create unique models of each system. A unique model means that the specific issues regarding the configuration can be studied, analysis of alternatives explored, and solutions defined. There are of course drawbacks to this approach:

- Stove-piped development of the systems means that lessons learned in one configuration cannot be easily distributed to others.
- Custom components for each system will increase supply chain costs and maintenance overheads.
- System development will take longer as each system will be developed from scratch rather than learning from previous ones.
- Increased project and product risk as unique configurations and components will require individual mitigations and resolutions.

MB-PLE provides solutions for these problems and more.
See [3], [4]

2. Product Line Engineering

Traditionally, product lines have evolved over time becoming increasingly difficult to manage and maintain. Manufacturers would create a single product for a specific purpose or customer. Variations of the product would be created when customers' needs changed or to improve production. Eventually, these would evolve into product lines. However, without a product line strategy, the number of variants overwhelms management and engineering. Product Line Engineering (PLE) was created to solve these problems.

PLE is the engineering and management of a group of related products using a shared set of assets and a means of designing and manufacturing. PLE can include both system and software assets and involves all aspects of engineering including electrical, electronic, mechanical, chemical, etc. As this whole of system approach is also essential for systems engineering, PLE is becoming more relevant to systems engineers. Model-Based Systems Engineering (MBSE) at the enterprise level using architecture frameworks such as the UAF [5] and the systems level using the Systems Modeling Language (SysML) [6], [12] is now becoming the norm in the industry. The recent International Council on Systems Engineering International Workshop and International Symposium contained many submissions on MBSE in a wide range of industries. This trend has been growing over the past 20 years and will continue to do so. PLE is being investigated particularly in the automotive arena, but also in rail, power systems, aerospace, manufacturing and MBSE in general. [7], [8]. These are all industries looking to adopt PLE and leverage its capabilities to achieve economies of scale and drive down product costs. Independent survey results have shown that applying MB-PLE approaches can reduce total development costs by 62% and deliver 23% more products on time. [7]

2.1. PLE Basics

A useful resource for information on PLE is ProductLineEngineering.com. It provides resources and information without trying to promote any specific tools. The resources page points to tools vendors for readers who wish to invest in tools. The text below in quotes is taken from that website. “Feature-based PLE is the modern digital engineering industry best practice for PLE, as defined in the INCOSE Product Line Engineering Primer, Feature-based Systems and Software Product Line Engineering: A Primer [9] and the ISO/IEC 26580 Feature-based PLE standard, Methods and tools for the feature-based approach to software and systems product line engineering [10]. Feature-based PLE offers significant improvements and benefits in effort, cost, time, scale, and quality by exploiting system similarity while providing formal variation/variant management.”

2.2. PLE and Models

Various tool vendors now provide PLE integrated with their MBSE environment. This paper examines the capabilities provided by the Dassault Cameo Enterprise Architect tool. PLE constructs can be applied at all levels of the architecture. An integrated approach means that the PLE feature set can directly reference the MBSE model. The PLE notation can be integrated into architecture frameworks such as the UAF and DoDAF [5, 11], systems architectures using the Systems Modeling Language (SysML) [6, 12].

Engineers can define a product line and its various options by defining a model called the 150% model. This contains the system along with all its possible system components, interfaces, behavior, requirements, etc. For example, this would define a car as simultaneously having a 4-, 6-, and 8-cylinder gasoline engine as well as a diesel engine. The PLE notation provides the ability to define a variation point of Engine, and then define that one and only one of the 4 engines above can exist in any actual product. In addition, dependencies between engine type and transmission type, exclusive or relationships, etc. can also be defined. Each of these components can be a complex system of systems in and of itself. Often the internal details of these systems are not pertinent or can increase the size of the model. In addition, it is often necessary to reuse the components without changing them. There may be several different versions of evolutions of the systems as well.

2.3. Why? Rather than What and How?

More important than the engine size and type, however, is the purpose of the vehicle using the engine. What will the vehicle be used for and in which environments? What loads does it have to support and for how long? Defining the product line based on the implementation features rather than the mission type, environment and collaborating systems will limit the useability of the product line definition as well as the resulting systems.

2.4. MB-PLE

Combining MBSE and PLE provides the ability to implement Model-based Product Line Engineering (MB-PLE) at all levels of architecture and throughout the various phases of the development cycle. Adopting an MB-PLE approach impacts the fundamentals of how organizations deliver and compete with their product lines. Adopting MB-PLE early in the development lifecycle allows the organization to capitalize on the advantages of MB-PLE and leverage the proven ROI of these techniques. It will also provide a decision framework to guide development and ensure the most appropriate product for the market, domain, and the customer. The variation points, variants, dependencies, and mutual exclusion constructs naturally lend themselves to the decision-making process as well as the product specification process. Using a decision execution engine, the engineer can review the results of the decision and perform trade-off analysis. The same techniques can be used for market analysis as well as detailed engineering making the techniques applicable for multiple stakeholders.

2.5. Shared Components

The PLE Primer [9] describes a third required aspect for PLE, which is a component library. Creating a common library of parts for the product line, is as practical as an auto mechanic keeping a common set of parts in the garage, or an engine factory keeping a parts database. Redefining and/or re-cataloging the parts for each engine would be time-consuming and error prone. Prior to starting the modeling exercise, it is important that the model structure as well as any model of model's structure be carefully considered before starting to reduce duplication.

3. The Example AT-AT Model

As noted above, the AT-AT is a key ground combat system for the Empire, and it is used in many of the conflicts it engages in as part of bringing order to the galaxy. It has several variants that it uses, which share a number of common sub-systems, but have different solutions within those sub-systems depending on the variant. The Empire wants to develop a library of system models for all AT-AT variants, but doesn't want to maintain them separately, and has opted to use PLE as a means of accomplishing this.

3.1. Initial Model Structure and Features

Since these system models were going to be used within other UAF mission models, the Empire used mission characteristics to define what variant would best be suited for the mission it was a part of. These characteristics, Environment and Mission Type, were then set as the defining features for AT-AT variant selection. This can be seen in Fig. 1, below.

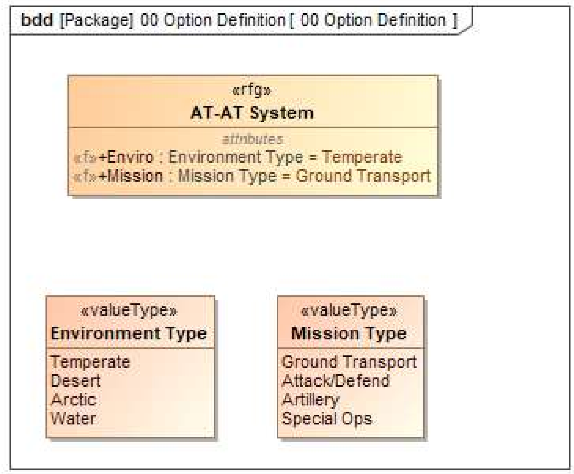

Figure 1: The AT-AT Variant Selection.

Next, the 150% system model (Fig. 2) was developed, showing all the possible sub-systems and components it could have, based on all the current AT-AT variants. This can include properties of those elements as well, such as weight, interfaces, and multiplicity, as well as any behaviors and flows between parts.

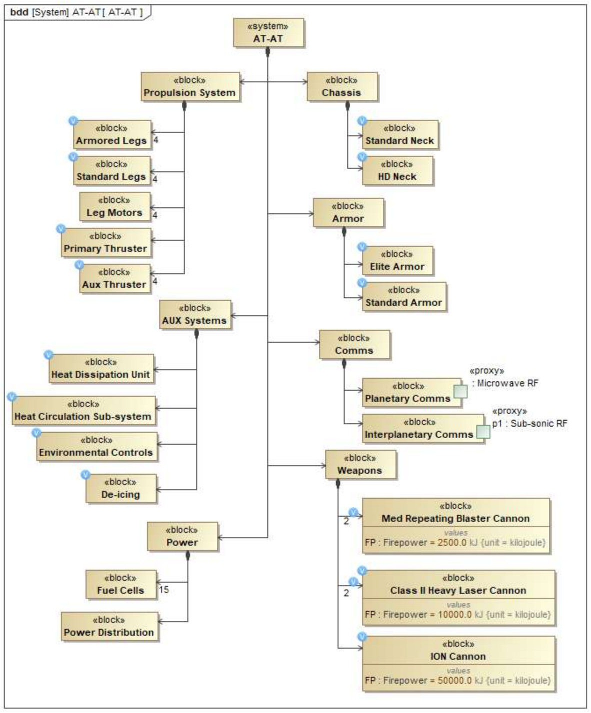

Figure 2: The AT-AT System Configuration.

3.2. Model Variation Points

Once the 150% system is defined, variation points can be added. In MD 2021x, you can add variation points that determine existence (which determines whether an element is present or not), or determine property values, all based on the features (<<f>>) defined within the Root Feature Group (<<RFG>>) or subordinate Feature Groups (<<FG>>). Fig. 3 shows a variation point definition for an element's existence that depends on the environment being selected as "Arctic".

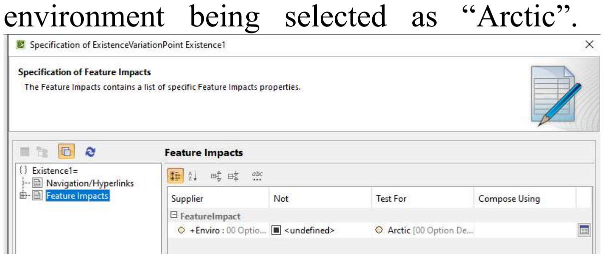

Figure 3: A Sample Variation Point Definition.

The MBPLE Plug-in for MD/CEA 2021x has some menus that enable simple variation point definitions. For more complicated definitions, such as property values, it requires some expression coding via the API (see Figs. 4 and 5, below).

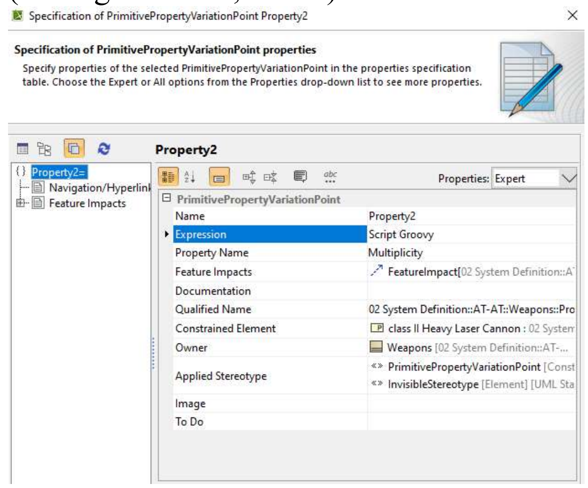

Figure 4: The Property Page Detail.

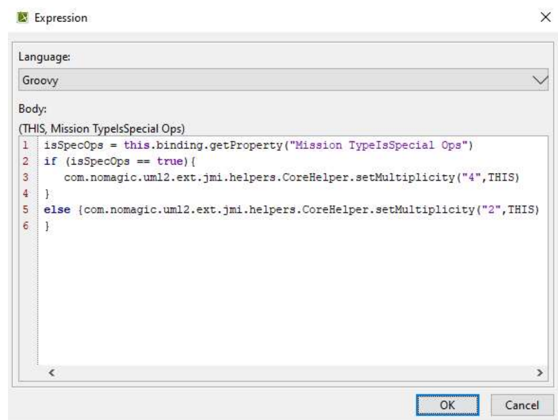

Figure 5: A Sample Script to Calculate Multiplicity.

3.3. Variants

With the variation points defined, the PLE model is now ready to have variants defined, including the variant to be used in the Battle of Hoth. In MD 2021x, the variations are simply instances of the <<RFG>>, with the features selected, as can be seen in Fig. 6.

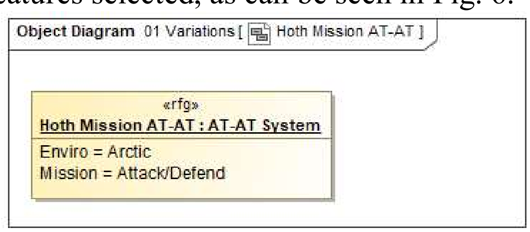

Figure 6: AT-AT Mission and Environment Selection.

The tool then allows you to apply a “Preview”, which allows you to select any variant you’ve defined, and it will highlight on diagrams any elements in your system that are included (green), not included (red), and changed (yellow). Note, changes are generally properties for which you’ve defined a variation point that changes the property value. The AT-AT system BDD in Fig. 7 is showing the Hoth variant preview.

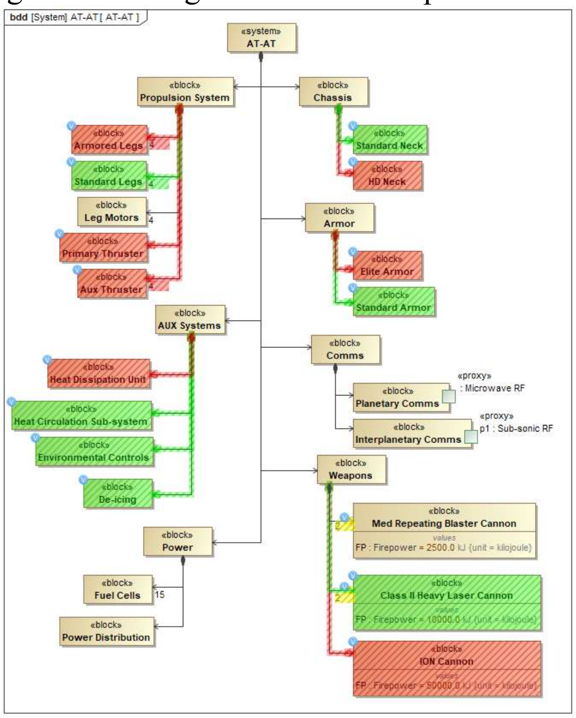

Figure 7: AT-AT Variant Selections Preview.

3.4. Re-usable System Model

Checking that the preview showed the correct configuration of the AT-AT for the Hoth mission, the next step was to produce a separate system model (not including the PLE info) that can then be used in other models. Once completed, the Arctic AT-AT system BDD looks like Fig. 8, with the highlighting now gone from the model, along with any of the elements that were previously highlighted red.

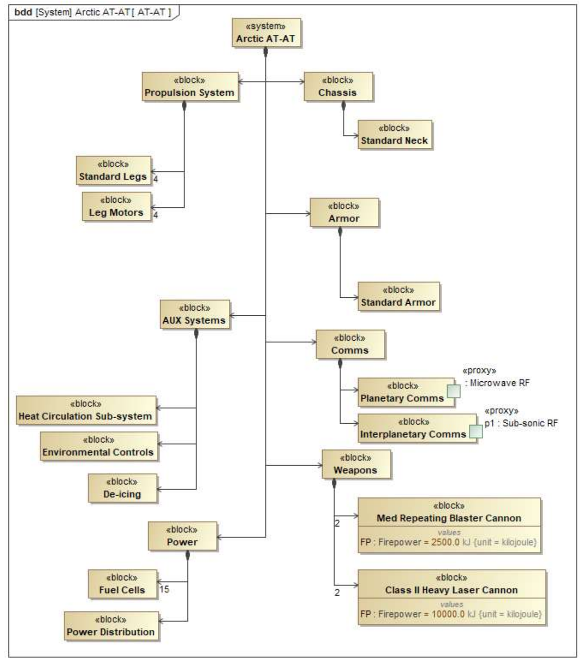

Figure 8: The Resulting AT-AT Configuration.

3.5. Integration into UAF Mission Model

The Hoth Battle Planners were eager to use this SysML model within their Hoth Mission model, and did so via MD 2021x Project Usage. Fig. 9 shows the overall model approach they took.

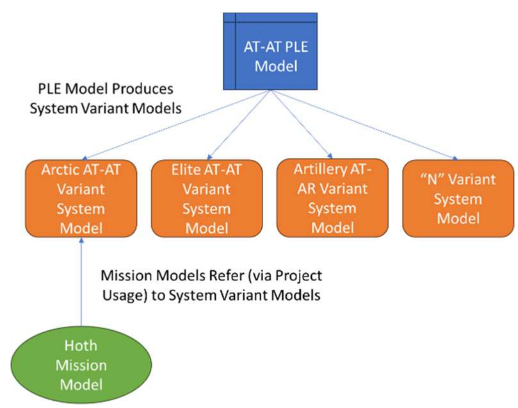

Figure 9: System of Systems Model Structure.

Since they already had some resource elements as placeholders in their mission model, they simply generalized the Arctic AT-AT element from the Arctic AT-AT system model, as shown in Fig. 10, below.

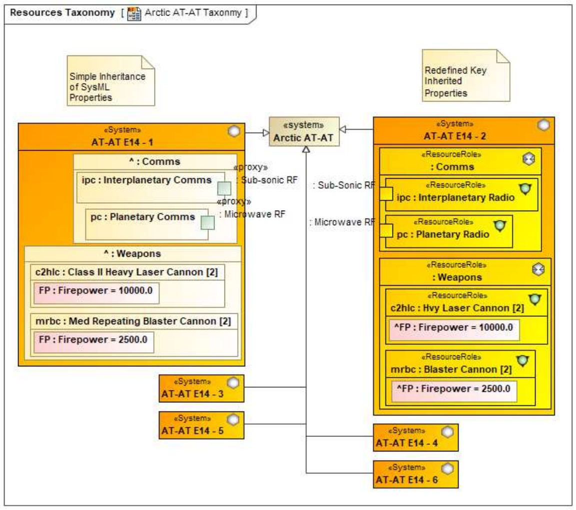

Figure 10: The Integrated AT-AT Model Configuration.

Note that each AT-AT in the mission structure is generalizing the Arctic AT-AT system element, and hence will inherit all its properties. Two AT-ATs are shown with some of their part and value properties to demonstrate this inheritance. Note that AT-AT E14-2 has had some of its properties redefined with the redefinitions being UAF elements. This redefinition may be needed if the mission model needs to show and track connections and flows between elements inherited from the SysML system model and other UAF elements in your mission model, as connectors cannot be drawn between resource ports (UAF) and proxy ports (SysML). It is recommended that prior to model integration that the amount of needed redefinition is well understood, as it could be a significant effort.

4. Reusable Libraries of Common Components

As you can likely see from the sections above, PLE provides an opportunity to develop a set of system libraries relatively quickly. It also offers the opportunity to control changes to a set of system models effectively by simply updating the PLE model, and then producing updated variant system models from that PLE model. This pattern can be used not only for system models, but mission models, process models, component models, etc. It's simply a matter of what kinds of models your organization finds value in re-use.

Additionally, having libraries of smaller building blocks (sub-systems, parts, etc.) can be used to help build the 150% system in the PLE model, allowing for easier and more controlled updates to the PLE (and hence variant system models) models. Having a documented and thought-out model library, curation, and configuration management plan is key to leveraging model libraries effectively. Systems can be assembled from existing parts, rather than rebuilt each time.

5. System Analysis and the Digital Thread

A major benefit of MBSE is the ability to use static and dynamic analysis techniques on the model. These include simulation of behavioral models, parametric analysis using system properties, and integration with external tools. Behavioral analysis includes execution of activity and state diagrams, both separately and simultaneously. Monte Carlo analysis of these behaviors can be run to determine average and optimal results of timing and overall execution. Parametric analysis can be run to review rollup of size, weight, and power (SWAP) and other analyses. Given the amount of energy used by the heavy LASER canons, it is essential that the power systems be adequately sized to last the entire battle. A group of systems can be analyzed to determine if combined fire power can be sufficient to break through force fields, etc. As UAF is built on SysML, these parametric diagrams can include both individual systems modeled in SysML as well as SoS elements in the UAF model. Finally, external physics-based tools can be used to perform mission analysis as discussed in [1].

6. Adopting a PLE Approach

Adopting an MB-PLE approach can be done for any organization regardless of its level of maturity and sophistication of MBSE. Adding MB-PLE to an ongoing project is normally ill-advised due to the disruption caused by additional learning curves for engineers and would likely cause unacceptable project delays. It is well known in the industry that project delays have strong and undesirable consequences for managers on Empire projects. However, an MB-PLE approach should be implemented at the end of virtually any project to harvest useful patterns, types, interfaces, components, and systems for use on future projects and to populate the component libraries.

7. Future Work and Conclusions

There are multiple directions in which this research could go, but there is one avenue that the authors are particularly interested in pursuing: use of MB-PLE with Mission Models. [MG2]As we've shown in this paper, PLE can be used to provide different system architectures from a single model. Similarly, PLE can be used in SoS models for mission engineering, allowing stakeholders to consider multiple resource architectures that are implementations of the operational architecture, and then produce individual mission models that can be used for additional analysis, or other uses.

This paper has shown the advantages of MB-PLE regarding land-based combat systems in a fictional environment. This was done to demonstrate ongoing work being accomplished on non-fictional systems. This can have a positive and lasting impact on the development of military systems and should be more widely implemented to leverage these techniques.

8. REFERENCES

[1] M. Gagliardi, M Hause., "Implementing Mission Engineering with UAF" In Proceedings of the Ground Vehicle Systems Engineering and Technology Symposium (GVSETS), NDIA, Novi, MI, Aug. 16-18, 2022.

[2] Battle of Hoth, Reference information Available online from https://starwars.fandom.com/wiki/Star_Wars:_Episode_V_The_Empire_Strikes_Back#The_Battle_of_Hoth

[3] Model-Based Product Line Engineering - Enabling Product Families with Variants, M. Hause, in Proceedings of the Ground Vehicle Systems Engineering and Technology Symposium (GVSETS), NDIA, Novi, MI, Aug. 12-14, 2015.

[4] M. Hause, A. Korff, Using Model-based Product Line Engineering for Decision Making and to Guide Product Development, In Proceedings of the Ground Vehicle Systems Engineering and Technology Symposium (GVSETS), NDIA, Novi, MI, Aug. 4-5, 2015.

[5] OMG, 2022, Unified Architecture Framework Specification Version 1.1. (2022). Retrieved 31 January 2022, from https://www.omg.org/spec/UAF/1.1/About-UAF/

[6] Object Management Group (OMG), 2019. OMG2012-06-01.OMG Systems Modeling Language (OMG SysML™), V1.6, http://www.omg.org/spec/SysML/1.6/PDF/.

[7] EMF 2013 Independent Survey Results from 667 Systems Engineering respondents, http://www.embeddedforecast.com/EMF_freewhitepapers43.php

[8] Product Line Engineering Success Stories. Available from https://productlineengineering.com/additional-resources/

[9] INCOSE PLE Working Group, Feature-based Systems and Software Product Line Engineering: A Primer (INCOSE, 2019)

Available from https://connect.incose.org/Pages/Product-Details.aspx?ProductCode=PLE_Primer_2019,

[10] ISO/IEC 26580:2021, Software and systems engineering. Methods and tools for the feature-based approach to software and systems product line engineering, available from https://www.iso.org/standard/43139.html

[11] DoDAF DoD CIO, 2012, DoD Architecture Framework Version 2.02, DoD Deputy Chief Information Officer, Available online at http://dodcio.defense.gov/dodaf20/dodaf20_pes.aspx, accessed June, 2014.

[12] Friedenthal, S., Moore, A., Steiner, R. Practical Guide to SysML: The Systems Modeling Language Second Edition, Morgan Kaufman, Oct 31, 2011