2023 NDIA MICHIGAN CHAPTER
GROUND VEHICLE SYSTEMS ENGINEERING
AND TECHNOLOGY SYMPOSIUM
DIGITAL ENGINEERING / SYSTEMS ENGINEERING TECHNICAL SESSION
AUGUST 15-17, 2023 - NOVI, MICHIGAN

PRODUCT LINE ENGINEERING WITHIN A DIGITAL ENGINEERING / MODEL-BASED SYSTEMS ENGINEERING ENVIRONMENT FOR GROUND VEHICLE SYSTEMS

Robert J. Kanon[1], Kevin W. Griffin[1], Surender K. Dua[1], Dennis J. Tuckowski Jr. [1], Michelle D. Gasbarro[1], Nathan D. Rojas[1], Steven Schultz[2]

SAIC[1], GVSC[2]
Sterling Heights, MI, Warren, MI

ABSTRACT

The ISO 26580 Software and systems engineering — Methods and tools for the feature-based approach to software and systems product line engineering (PLE)[1] standard provides the guidance and direction to be successfully applied to military ground vehicle systems. Implementing PLE provides military stakeholder benefits such as increased commonality, functionality, quality, and productivity, while also delivering decreased risk and cost. Moreover, these benefits last through multiple generations of ground vehicle sytems. Merging PLE into a Digital Engineering (DE)/Model-based System Engineering (MBSE) environment is a critical proven combination for successful automotive manufacturers today. This same approach can be extended to the military ground vehicle family of vehicles resulting in the ability to rapidly design, assess, evaluate, simulate, manage, and instantiate product variants from a common feature catalog.

1. INTRODUCTION

Traditional ground vehicle programs often follow a normal product-centric approach, that when a new system is needed which resembles other systems that already exist, engineers select the most similar one available, copy it, make changes to it, and field it as the starting point for the new system. This is referred to as the clone-and-own approach. This helps the new program by benefitting from the successes of the first program, at least initially. Once the new program proceeds, then both programs encounter their own challenges, constraints, test issues, and more as they travel down the rest of the product lifecycle. As more programs and variations within a program come on-line, then this same approach means that multiple teams often encounter, analyze, re-engineer, and solve the same issues as their fellow teams, resulting in wasted and duplicitous efforts. When a common component exists on several programs, it would be ideal to design, test, implement, and maintain it once by a single team, and then make it available to all programs to use this particular component.

This is the challenge that Product Line Engineering addresses well, and it has been applied across automotive, aerospace, and numerous other product industries efficiently with great benefits. Essentially PLE allows a team to engineer a portfolio of related products, taking advantage of commonality and products' similarities, and efficiently dealing with the differences. All lifecycle engineering disciplines are still involved across the product lifecycle, from planning and requirements to production, sustainment, and obsolescence.

Rather than have numerous but similar separate vehicle system programs, PLE reflects the product line portfolio as a single entity to be managed.

Ground vehicle systems need to support varying missions, operating environments, customer needs, and regulatory frameworks – all while providing safety and reliability. To this, add-in additional program needs for commonality, Modular Open Systems Approach (MOSA), ease of integration of future technology, and a migration to digital engineering, and you have a very complex family of vehicles, all which contain a core set of functionality and design components the same or similar to other vehicle systems. PLE represents a cost-effective and cost-efficient way to manage variations while ensuring quality and responsiveness to these program objectives.

Figure 1 - DoD Digital Engineering Strategy Goals

PLE is consistent with the Department of Defense Digital Engineering Strategy [2] as it covers the first three goals (utilization of models, providing an authoritative source of truth, and incorporating technological innovation), as seen in Figure 1. The ISO 26580 reference model provides an infrastructure as per goal 4, as well as starting the path toward mentorship of the workforce culture (goal 5). Implementing a PLE approach inside MBSE and extending it to a full digital engineering environment is consistent with Army goals to improve how it executes institutional processes such as requirements development, acquisition, planning, programming, budgeting, and execution (PPBE).[3] Successful adoption of PLE impacts every single area along the product lifecycle and has demonstrated benefits and value in automotive and commercial industries.

2. ISO 26580

ISO 26580 Software and systems engineering — Methods and tools for the feature-based approach to software and systems product line engineering (PLE) is a relatively new standard released by the International Organization for Standardization (ISO). The ISO 26580 provides a framework, guidance, and direction to implement product line engineering on ground vehicle programs. The standard elicits a specific and repeatable approach to PLE that takes advantage of commercial off-the-shelf industrial-strength tools and technology which provides the detail for future tools to develop compliant capabilities. The usage of a feature-based PLE methodology means less upfront analysis, design, and implementation effort is required prior to gaining the benefits from the approach.

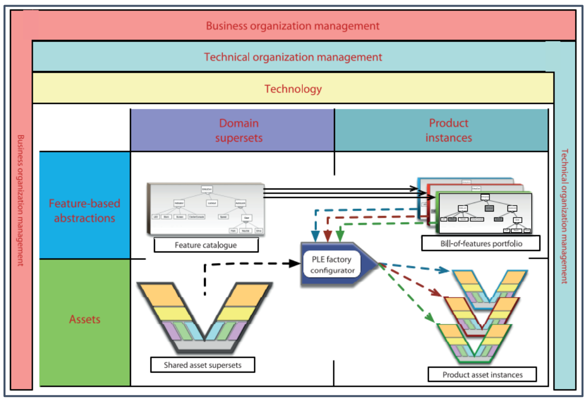


Figure 2 – ISO 26580 Reference model for feature-based product line engineering

The ISO 26580 standard provides a reference model consisting of an abstract representation of the key technical elements, tools, and methods of feature-based PLE, as shown in Figure 2. Tailoring a best practice PLE approach for ground vehicle systems can and should be traced back to the standard to fully realize the benefits and value that application PLE will bring. Major concepts that are vital to efficiently applying a feature-based PLE methodology:

a) a mapping from features to asset variation points
b) a shift of all design and implementation effort
c) a single product instance (or variant) is achieved via automated configuration of member product instances

3. IMPLEMENTING PLE INSIDE MBSE

Past efforts to adopt a PLE approach tended to involve using PLE-specific tooling that needed adapters to interact with every other tool along the product lifecycle. This approach meant having special point-to-point tool interfaces, learning curve challenges, a great deal of patience, and regular involvement and consulting from the tool vendors to make this work successfully.

Now, with the ISO 26580 standard in place, other tool vendors can create standard compliant capabilities to their tools to permit an easier adoption of PLE within their MBSE environment.

System engineers can now use Systems Modeling Language (SysML) to capture the product variation details and constraints across the entire family of products in order to create any specific product.

The usage of SysML is gaining momentum within the Department of the Army as the focus on open architectures, scalability of technology, and rapid future integration are key points of the Army's modernization strategy. SysML models of a platform provide a way to represent the relationships and design of the interfaces and growing complexities of modern combat systems.

A typical starting point is to capture the details of variation that go into the family of products. To accomplish this, we construct a feature model which provides the details for what is available for downstream selections in a specific vehicle variant. A feature catalog is a model of the collection of all the feature options and feature constraints available across the product line.

Next, create your requirements and system architecture models in a similar fashion as before. A single major difference is that we are now creating what is referred to as a 150% model. All the requirements and features across the entire family of vehicles will be present in these models, but not every requirement or system block will be used on a single vehicle. Each vehicle variant will exist through identification of the specific selections of what to include and exclude from the 150% model.

A key ability to execution of PLE within an MBSE environment is the ability to identify and create variation points in the 150% models. These variation points provide the constraint expression necessary to be able to select specific variant options and combinations, and are critical for the automation in the creation of specific vehicle variants later in the process. The variation point is described in a SysML constraint expression, which can then be applied against other modeling artifacts like requirements, blocks, state machine states and transitions, and many more.

With the prior process steps in place, then the creation of specific vehicle variants becomes fairly simple. The creation of a variant is very similar to the experience of purchasing a new automobile of today – you may intend to purchase a mid-size sedan, but then your salesperson will ask which of three engines you would prefer, and did you want cloth or leather interior, an upgraded audio system, etc. The same methodology is now employed in selecting options from the feature catalog to create a specific instance from all the available options for your family of vehicles.

There exists direct applicable to this approach for defense systems. As defense systems begin to leverage and integrate common communications devices, lethality systems, active protection, sensors, and video displays across platforms, the benefits of this approach will create a more re-usable method for these common integrations that take advantage of common architectures.

This instance (or vehicle variant) is then viewable by automation in the SysML modeling tool. Cameo permits a capability called previewing variants, and through a single pulldown selection, you can select a preview. Then, navigating to various diagrams and tables will show you those portions of the model that are "included" in the instance, and conversely, also indicate the portions that are excluded based on the specific instance selection. Figures 3-4 show included and excluded blocks and requirements for a particular selected variant preview. Artifacts highlighted in green are included, and those in red are excluded for this particular vehicle system variant.

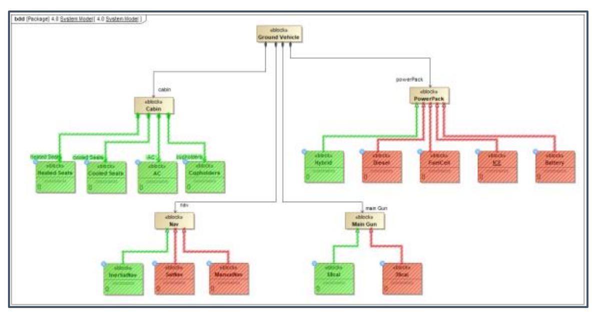
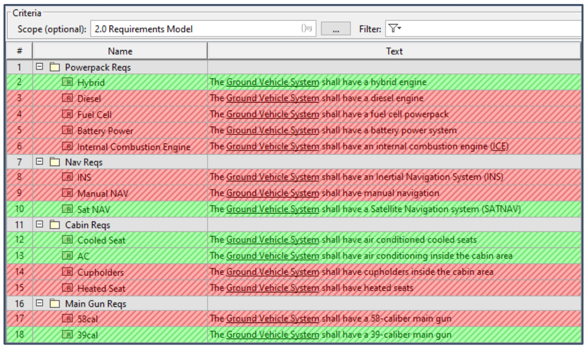


Figure 3 - Example Block Definition Diagram Preview



Criteria

Scope (optional): 2.0 Requirements Model ... Filter:

#	Name	Text
1	Powerpack Reqs	
2	Hybrid	The Ground Vehicle System shall have a hybrid engine
3	Diesel	The Ground Vehicle System shall have a diesel engine
4	Fuel Cell	The Ground Vehicle System shall have a fuel cell powerpack
5	Battery Power	The Ground Vehicle System shall have a battery power system
6	Internal Combustion Engine	The Ground Vehicle System shall have an internal combustion engine (ICE)
7	Nav Reqs	
8	INS	The Ground Vehicle System shall have an Inertial Navigation System (INS)
9	Manual NAV	The Ground Vehicle System shall have manual navigation
10	Sat NAV	The Ground Vehicle System shall have a Satellite Navigation system (SATNAV)
11	Cabin Reqs	
12	Cooled Seat	The Ground Vehicle System shall have air conditioned cooled seats
13	AC	The Ground Vehicle System shall have air conditioning inside the cabin area
14	Cupholders	The Ground Vehicle System shall have cupholders inside the cabin area
15	Heated Seat	The Ground Vehicle System shall have heated seats
16	Main Gun Reqs	
17	58cal	The Ground Vehicle System shall have a 58-caliber main gun
18	39cal	The Ground Vehicle System shall have a 39-caliber main gun

Figure 4 - Example Requirements Table Preview

The application of variation points can extend beyond requirements tables and block definition diagrams (BDDs). Internal block diagrams, state machine diagrams, activity diagrams, and even Graphical User Interface (GUIs) diagrams (used during simulation) can all have variation points. When in the preview mode, you can see which portions are included and excluded for a specific variant. An example showing the preview for inclusion (or exclusion) of a automatic sensor in both an internal block diagram and the state machine behavior is shown in Figure 5.

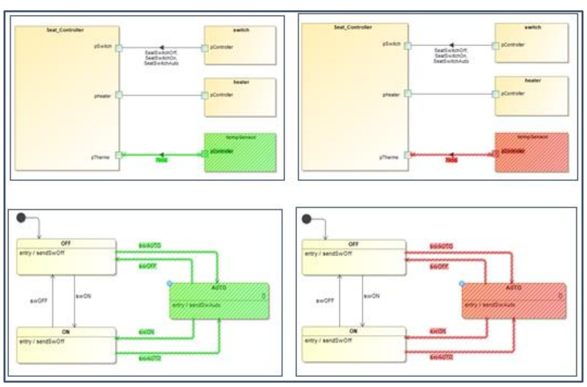


Figure 5 - Example IBD and State Machines

To meet major milestones and releases for a specific variant program, there exists a capability to create a full model transformation. This involves the complete removal of those parts of the 150% model that are not included in the specific variant. Care should be taken to save the transformed model to a separate location to not damage the 150% model. Process steps and modeling governance can ensure these steps are followed properly.

4. NOTIONAL GROUND VEHICLE CONCEPT DEVELOPMENT EXAMPLE

A simple notional ground vehicle system example is presented to walk through the main steps of the process implementing PLE across the requirements and system architecture engineering disciplines.

Step 1 – Develop the feature model. Our family of ground vehicles shall have a cabin which will have further customizable content, one of 5 possible engines, one of 3 possible navigation systems, and one of 2 possible main guns. Our top-level feature model is shown in Figure 6.

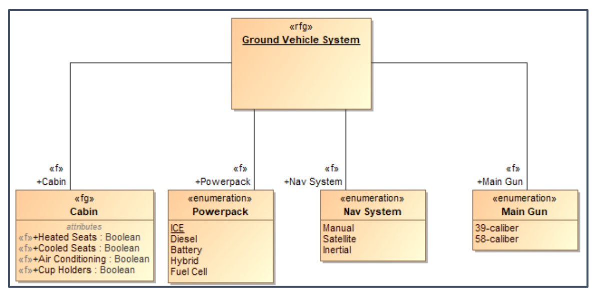


Figure 6 - Feature model

The cabin feature will have another layer of feature choices, and it is captured in Figure 7. Since the choices inside the cabin are all simple include/exclude (Boolean) options, then they can be shown inside the cabin feature group block.

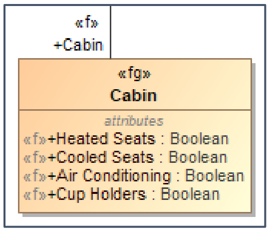


Figure 7 - Second layer feature details

Step 2 – Create requirements model. A requirements table was created to easily view the various requirements for our family of ground vehicles, and it appears in Figure 8. As can be seen with a quick inspection, no single vehicle is going to have all these requirements – as this is a 150% model, and the requirements are created and maintained for all vehicles. For a specific variant, only a subset of these requirements will be applicable. The variation points that will be added will be utilized in the automation to include or exclude a requirement.

Criteria
Scope (optional): 2.0 Requirements Model   Filter:

#	Name	Text
1	Powerpack Reqs	
2	Hybrid	The Ground Vehicle System shall have a hybrid engine
3	Diesel	The Ground Vehicle System shall have a diesel engine
4	Fuel Cell	The Ground Vehicle System shall have a fuel cell powerpack
5	Battery Power	The Ground Vehicle System shall have a battery power system
6	Internal Combustion Engine	The Ground Vehicle System shall have an internal combustion engine (ICE)
7	Nav Reqs	
8	INS	The Ground Vehicle System shall have an Inertial Navigation System (INS)
9	Manual NAV	The Ground Vehicle System shall have manual navigation
10	Sat NAV	The Ground Vehicle System shall have a Satellite Navigation system (SATNAV)
11	Cabin Reqs	
12	Cooled Seat	The Ground Vehicle System shall have air conditioned cooled seats
13	AC	The Ground Vehicle System shall have air conditioning inside the cabin area
14	Cupholders	The Ground Vehicle System shall have cupholders inside the cabin area
15	Heated Seat	The Ground Vehicle System shall have heated seats
16	Main Gun Reqs	
17	58cal	The Ground Vehicle System shall have a 58-caliber main gun
18	39cal	The Ground Vehicle System shall have a 39-caliber main gun

Figure 8 - Requirements table for 150% model

Step 3 – Create system model. A notional block definition diagram in Figure 9 shows the main blocks of our example, and their relationships. This diagram utilizes the usual artifacts in creating a system model, and all typical SysML relationships – therefore, this step will seem very similar for our system modeler to complete.

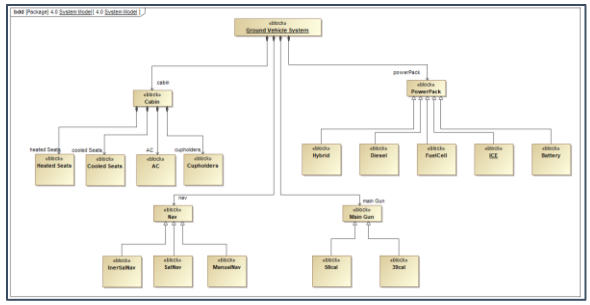


Figure 9 - System Block Definition Diagram

Step 4 – Create variation constraints. We first create a SysML constraint and stereotype it as an existenceVariationPoint. Then, the feature impacts are defined for the variation point. The completed HeatedSeat constraint is shown in Figure 10, and shows it is applicable to the Boolean option of whether Heated Seats exist.

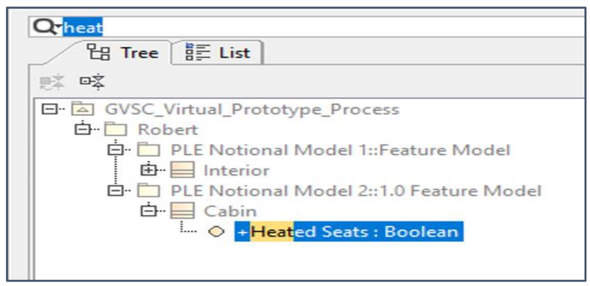


Figure 10 - Example Variant Point Constraint definition for Boolean feature option

Another example is shown in Figure 11, where one option in a list of choices can be selected. For this option, the constraint will impact a feature when the Powerpack is selected as Hybrid, which is defined in the "Test For" field.

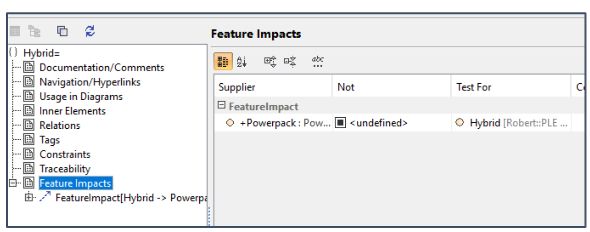


Figure 11 - Example Variant Point Constraint definition for Enumeration feature option

Once the variation constraints are completed, then they need to be applied to the system model elements and requirements. This is accomplished via drag and drop functionality and verified via a new "blue V" icon in the upper right-hand corner of the element image in the browser, as shown in Figure 12.

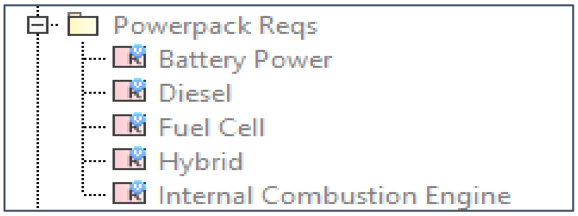


Figure 12 - Variation Point icons

Step 5 – Create and preview variants. Creation of the specific vehicle variants can be easily accomplished inside a generic tabular view. Names can be assigned to each variant, and selections made from each feature or feature group, as shown in Figure 13.

#	Name	Main Gun : Main Gun	Powerpack : Powerpack	Nav System : Nav System	Cabin : Cabin
1	Base Ground Vehicle	39-caliber	Diesel	Manual	base cabin : Cabin
2	Ground Vehicle 1	58-caliber	ICE	Satellite	cabin cold weather : Cabin
3	Ground Vehicle 2	58-caliber	Hybrid	Inertial	cabin luxury : Cabin
4	Ground Vehicle 3	58-caliber	Battery	Satellite	cabin luxury : Cabin
5	Ground Vehicle 4	39-caliber	Hybrid	Satellite	cabin hot weather : Cabin

Figure 13 - Product Line Variant selections in a table

Now, it is simple to visualize a single vehicle variant at any time in the development process. A simple pulldown menu permits a selection from any of the defined variants, as shown in Figure 14. Once in preview mode, then all diagrams will be display color annotations to indicate which model elements are included (in green), and which model elements are excluded (in red). See Figures 1-3 for examples with included and excluded modeling elements for a specific variant preview.

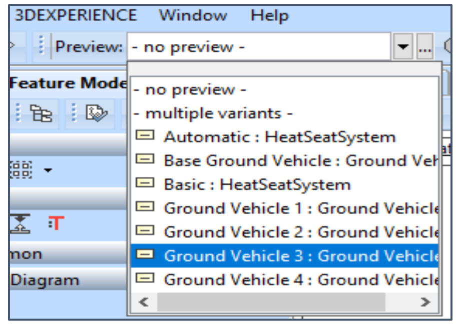


Figure 14 - Preview pulldown menu selection

5. EXTENSION TO A DIGITAL ENGINEERING ECOSYSTEM

So far, the extent of the PLE has been within the confines of the MBSE tool. Having a collaborative integrated digital engineering ecosystem, such as SAIC's ReadyOne, will permit the integration of all key data into the DE ecosystem. The data model, business rules, and governance needs to be developed to capture not only the 150% system model structure, but also each variant model structure and associated key artifacts. The system and PLE data can then have relationships to downstream product engineering teams such as advanced simulation, design, testing, production, and operations. An example screenshot in Figure 15 shows the relationships and traceability between the system elements and other artifacts from various engineering disciplines and tools within the ReadyOne environment.

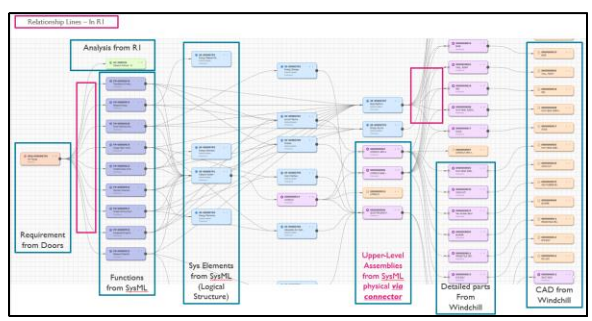


Figure 15 - DE digital thread : connected data in semantic web within ReadyOne

Variation management capabilities inside the ReadyOne product line management solution can now extend into other engineering disciplines through these data relationships, focused workflows, and automation, and enable direct links back to the authoritative source of truth in our MBSE/PLE model. This type of solution achieves the DoD goals for Digital Engineering Transformation.

6. CULTURAL CHANGE

As the Army's future systems continue to modernize and integrate the latest and greatest technologies, the needs for Digital Engineering approaches such as Product Line Engineering are emerging as ways to manage the growing complexities and speeds at which we need to operate within. This transformation to digital engineering and digital business processes will require highly skilled and talented leaders and decision makers to interpret, understand, and be comfortable working with MBSE models. This also needs to be supported by highly trained SysML and Digital Engineering professionals who can prepare and make sense of the growing amount of data being made available. Leadership teams also need to embrace and plan for PLE, with visions across programs looking to capture future needs and capabilities. The potential for the DoD to be able to greater understand ground system design decisions and available interfaces will speed integration. Digital approaches also have the potential to save cost and time by bringing all stakeholders into the design and decision processes earliest to ensure that all risks are identified and inform design requirements. SAIC has long been a strategic partner of GVSC and is preparing and investing in tools to support the Army's future modernization initiatives and ways to successful achieve the vision for an Army of 2040.

7. CONCLUSIONS

Military ground vehicle systems can realize the benefits provided by the implementation of product line engineering within the model-based system engineering discipline by adopting and tailoring this approach built upon the ISO 26580 standard. Following these steps, a ground vehicle program can accelerate their digital transformation and be strategically prepared to better define and achieve increased commonality, functionality, quality, and productivity across the family of vehicles, while also delivering decreased risk and cost.

8. REFERENCES

[1] ISO 26580 Software and systems engineering — Methods and tools for the feature-based approach to software and systems product line engineering (PLE), April 2021.

[2] Department of Defense, "Digital Engineering Strategy," Office of the Deputy Assistant Secretary of Defense for Systems Engineering, Washington, D.C., June 2018.

[3] Office of the Army Chief Information Officer, "Army Digital Transformation Strategy", Headquarters, Department of the Army, October 2021.