Realizing Training Transformation Through Feature-based Product Line Engineering

Brett Tainter
The Boeing Company
St. Louis, MO
brett.tainter@boeing.com

Randy Pitz
BigLever Software
St. Louis, MO
rpitz@biglever.com

ABSTRACT

Systems and Software Product Line Engineering (PLE) has gone mainstream in complex engineering environments and is being increasingly adopted by many businesses. We will provide an overview of the recently released ISO 26580: “Methods and Tools for the Feature-based Approach to Systems and Software Product Line Engineering” which is an automated approach for managing a product portfolio as a single entity with variation as opposed to products developed in isolation. There are numerous references that ascribe cost and efficiency benefits to organizations adopting a PLE approach, especially in the Aerospace and Defense sector.

However, realizing PLE adoption within simulation-based training systems has lagged other defense systems despite similar system complexity and reduced regulatory oversight.

The advantages of the PLE methodology includes documented benefits ranging from acquisition through development and sustainment. These benefits are no less relevant to the modeling, simulation and training community than the systems they are modeled after. The results achieved through this methodology can transform training by enabling more innovation throughout the development life-cycle.

The P-8 Mission Trainer architecture team is modernizing their development approach through use of PLE and Digital Engineering techniques to establish a single source of truth in a product line context. This approach is key to enabling the scalability of the product line as additional programs are added to the portfolio. The benefits are seen throughout the development life-cycle, across the concurrent baselines and is driving a culture focused on innovation. In this paper, we explore the benefits of PLE to large-scale training systems by primarily examining lessons learned from the Boeing P-8 training system architecture team along with supporting lessons found in other similar scale training system product lines.

ABOUT THE AUTHORS

Brett Tainter is the Model-Based Systems Engineering (MBSE) and Product Line Engineering (PLE) lead engineer for the Government Training Systems and Services organization within The Boeing Company. He has 15 years of experience developing training systems for several platforms as both a software engineer and system architect. During this time, he spent his first 12 years focused on the development of reusable Instructor Operation Station software. In his current role as the lead engineer for MBSE and PLE, he is responsible for guiding the digital transformation to these training systems. He is an active participant in the Simulation Interoperability Standards Organization (SISO) Distributed Interactive Simulation (DIS) working groups primarily focused on Simulation Management. He holds a Bachelor's degree in computer engineering from the University of Missouri-Columbia.

Randy Pitz is the Vice President of Customer Success at BigLever where he facilitates education, innovation and adoption of PLE through the onePLE Studio customer experience center. He is responsible for collaborating with customers to develop specific strategies to leverage the benefits of PLE. He has over 29 years’ experience in systems and software architecture and was a Technical Fellow for The Boeing Company where he architected product lines for simulation, safety-critical aviation and mission-critical avionics systems. Randy holds a Masters in Computer Science from Washington University, and his Bachelor of Science from Michigan Technological University.

Realizing Training Transformation Through Feature-based Product Line Engineering

Brett Tainter
The Boeing Company
St. Louis, MO
brett.tainter@boeing.com

Randy Pitz
BigLever Software
St. Louis, MO
rpitz@biglever.com

INTRODUCTION

The U.S. Navy's P-8A training system, developed by The Boeing Company as prime contractor, provides aircrew and maintenance training with realistic simulation enabling mission-ready crews to complete 70 percent of required training from this ground-based solution. The system includes courseware, software, equipment, personnel and logistical support to achieve this objective. Refer to Figure 1 for representations of the Aircrew Training Devices which will be the focus of this paper. Since the first delivery to the USN in 2011, variations of this system have been delivered to foreign partners that include U.K., Australia, New Zealand, South Korea, India and others in consideration. The growth of these opportunities has created a need to efficiently manage the system variations within a continuously changing engineering environment.

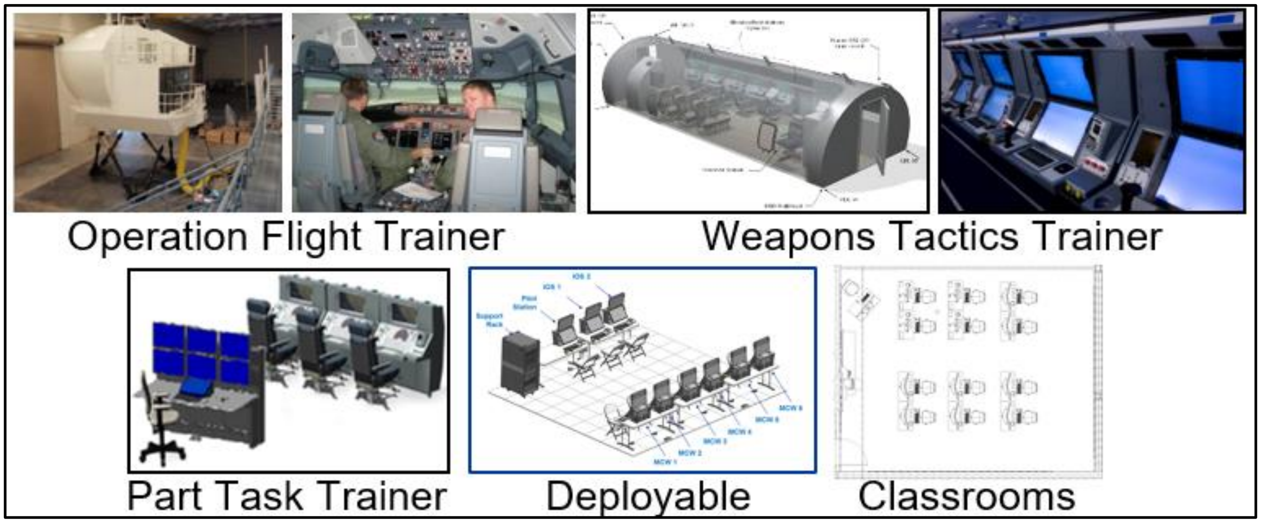


Figure 1 - P-8 Aircrew Training Devices

In 2017, Boeing Program Management realized that supporting multiple P-8A customers could not scale using the traditional isolated program silo approach. Two key Digital Engineering initiatives were kicked off in 2017 and 2018 for Model-Based System Engineering and Product Line Engineering. These Digital Engineering initiatives have been transforming the engineering approach of how this comprehensive and complex system is developed and maintained. A shift to formally managing the differences between the system instantiations unlocks key efficiencies unobtainable through traditional isolated development activity. These differences are often relegated to tribal knowledge and can be lost through staff turnover.

Model-Based Systems Engineering (MBSE) is a systems engineering approach using models, which are digital abstractions of the system design, as the source of truth for engineering activity. The models include requirements, structure and behavior design, and provide semantic formality that enable analysis, traceability and earlier detection of errors. The International Council on System Engineering (INCOSE) defines MBSE as "the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases." (INCOSE 2007)

Product Line Engineering (PLE) is an approach to efficiently develop reusable engineering assets. PLE has been around for many decades with many success stories (Lanman et al., 2011; Clements et al., 2014; Clements, 2015; Young et al., 2017; Teaff, 2019; Gregg et al., 2020; SPLC Product Line Hall of Fame). Feature-based PLE extends the concepts of first generation PLE and these examples demonstrate the growing success over the past decade. This modernized approach to PLE is being used to develop some of the most complex systems in Aerospace & Defense, Automotive and Software organizations. Therefore, we conclude Feature-based PLE has become mainstream based upon these published examples. Despite these successes, except for the U.S. Army Live Training Transformation (LT2) program and the General Dynamics Consolidated Product-line Management solution (Lanman et al., 2011), we are not aware of any other published examples describing PLE success in the simulation-based training domain.

In this paper, we aim to describe how the P-8A training system has been applying MBSE and PLE together in a novel approach called Product Line Modular Models (PLMM) to transform training for this remarkable platform. This innovative large-scale simulation-based training system reuses models across systems of different fidelity and complexity within one product line that includes classroom, part-task, crew and collaborative distributed simulation exercises. The P-8A training system product line provides ample opportunities to leverage sharing across the system family, and adjacent product lines, exemplifying what can be achieved through modern digital engineering.

FEATURE-BASED PRODUCT LINE ENGINEERING

Earlier approaches to PLE almost all emphasized a strong dichotomy between domain engineering and application engineering, sometimes called core asset development and product development (Clements & Northrop, 2002; Linden, et al., 2007; Krueger & Clements, 2013). Application engineering involved choosing reusable domain-engineered core assets and grooming them for use in a product as needed. By contrast, Feature-based PLE was introduced in the early 2000's as a specific form of PLE which generates product asset instances through automation and essentially eliminates application engineering. The innovations introduced with Feature-based PLE have been widely practiced and have demonstrated greater benefits than the early generation PLE techniques.

Feature-based Product Line Engineering: ISO 26580

In April 2021, a new standard was released that codifies the practice known as Feature-based Product Line Engineering (PLE Primer, 2019), also sometimes referred to as second generation PLE. That specification, ISO/IEC 26580 Software and Systems Engineering – Methods and Tools for the Feature-based Approach to Software and Systems Product Line Engineering, is what we will briefly summarize here.

Reference Model (Figure 2)

- **Feature Catalogue**: captures the features that are available for each product to select. A feature is a distinguishing characteristic that describes how the members of the product line differ from each other. This provides a common language and definition of the product line's scope of variation for everyone throughout the organization.
- **Bill-of-Features**: captures the product configurations that are supported by the product line.
- **Shared Asset Supersets**: These are the engineering artifacts that support the creation, design, implementation, deployment and operation of member products. These artifacts are anything represented digitally such as requirements, design models, source code, test cases, Bill-of-Materials, wiring diagrams, documents, user manuals and more. As supersets these assets contain all engineering content that are needed by any product asset instance. The traceability between superset assets is developed and maintained throughout the life-cycle.
- **PLE Factory Configurator**: a piece of technology that is responsible for automatically generating product asset instance data by applying a Bill-of-Features to the shared asset supersets.

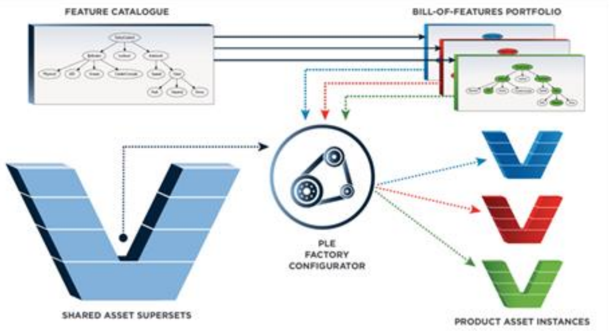


Figure 2 - The PLE Factory

- **Product Asset Instances**: the output from the PLE Factory that relates to one distinct member product from the product line. These are the set of artifacts that are used by engineers to produce the product. Engineers work on the feature catalog, bill-of-features and shared asset supersets through well-defined governance processes.

The ISO 26580 standard additionally defines people, roles and processes to operate a PLE Factory as defined in the reference model above. These are broken into two categories:

- Technical Organization Management Layer – people, roles and processes that relate to the engineering and technical management of the reference model elements.
- Business Organization Management Layer – people, roles and processes that relate to healthy adoption and execution.

Once the PLE Factory is established, product specific engineering assets are instantiated from the shared asset supersets rather than manually created and maintained. The task of engineering a portfolio of products is transformed through Feature-based PLE into the much more efficient task of engineering a single system: The PLE Factory itself.

PLE for Large-Scale Training Systems

What do the variations captured in a Feature Catalogue for a Training System look like? Any meaningful simulation will have parity with the real-world system that is being modeled in terms of the platform and mission capability. As those capabilities change across system instantiations, they are captured as differences that are modelled in the Feature Catalogue. Examples of these differences include radios, sensors, weapons and mission computing capability.

Additional sources of variation come from the simulation systems used to model real-world effects such as environmental servers, threat servers and visual systems. Also, there are the systems needed to plan, execute and debrief the training exercises. Sometimes real platform hardware or software is utilized in a trainer to achieve higher fidelity training. Finally, there are the computing and networking systems used to host these simulation capabilities and provide connectivity to support classroom, student, crew and distributed events. All these capabilities specific to the simulation devices may vary from system to system according to customer unique requirements. The combinations of platform and simulation capability variations create complexity that Feature-based PLE is ideally suited to formally manage.

Given all the sources of variability that exist within training system product lines we expected there would be many case-studies and success stories available. However, we find more examples of PLE success from complex safety-critical and mission-critical real-world systems that are based on DO-178C, MIL-STD-882E or ISO 26262. This seems counter-intuitive and implies the need to accelerate training transformation across training system solutions.

The organizations that figure out how to manage the complexity driven by variation within their product lines will have the ability to be more competitive and innovate. These benefits come from avoiding unnecessary duplication of effort and overhead of keeping multiple baselines synchronized that is achieved through Feature-based PLE. Many additional strategies for handling important considerations to Aerospace and Defense have been discussed by other publications (Clements, et al., 2013; Clements, et al., 2014; Gregg et al., 2020;).

DIGITAL ENGINEERING

The *Department of Defense (DoD) Digital Engineering Strategy (2018)* defines Digital Engineering (DE) as “an integrated digital approach that uses authoritative sources of systems' data and models as a continuum across disciplines to support lifecycle activities from concept through disposal.” Additionally, the vision describes a key benefit of DE is to “enable the use of models throughout the lifecycle to digitally represent the system of interest (i.e., system of systems, systems, processes, equipment, products, parts) in the virtual world.”

The Digital Engineering approach described in this paper expands on the typical single system focus to that of building a system family. Krueger and Boucher (2022) describe elevating systems engineering practice to system family engineering by adding PLE. This becomes a digital engineering approach that includes the management of variation driven by the proliferation of customer preferences and resultant complexity. Delligatti (2014), explains MBSE is built upon the three key pillars modeling language, tool and method. The MBSE selected method, such as Object-Oriented Systems Engineering Method (OOSEM), Harmony or MagicGrid, is used keep the models relevant throughout the system or product lifecycle. The formalized practices of MBSE and PLE are both leveraged as digital models used throughout the lifecycle to manage the entire system family.

MBSE is not a new type of engineering, instead it prescribes how to do systems engineering using a formalized process to ensure the appropriate level of rigor is applied. The MBSE model is the Authoritative Source of Truth (ASoT) for the design of the system and the processes and tools used to design the system. In order to do effective MBSE, organizations need guidance including how to apply the language, how to apply the framework and other customer specific modeling customizations. As the modeling process matures, additional oversight is needed to ensure model correctness and completeness, ideally through automation. Many of these processes and automation capabilities can be contained within the MBSE environment to provide engineers both the guidance and automated model validation needed to ensure correctness and completeness of the MBSE models.

As organizations grow and these processes are shared across products, tailoring though PLE becomes a necessity. To ensure the PLE variation is applicable across the lifecycle, the PLE models of variation are intentionally distinct and orthogonal to the MBSE models being developed. Applying PLE to the systems engineering efforts provides a significant advantage as it spans such a long time period over the product's lifecycle.

P-8A TRAINER BACKGROUND

The P-8 training systems are organized in three main categories focused on training the aircrew, operators and maintenance crew. Within each of these categories there are multiple training device types to align to the various stages of training starting with individual training, progressing to collaborative part task training and finally full mission training. The remainder of the paper will focus on the operator training devices, but the concepts apply to all categories. Figure 3 shows a conceptual architecture superset for the operator training devices.

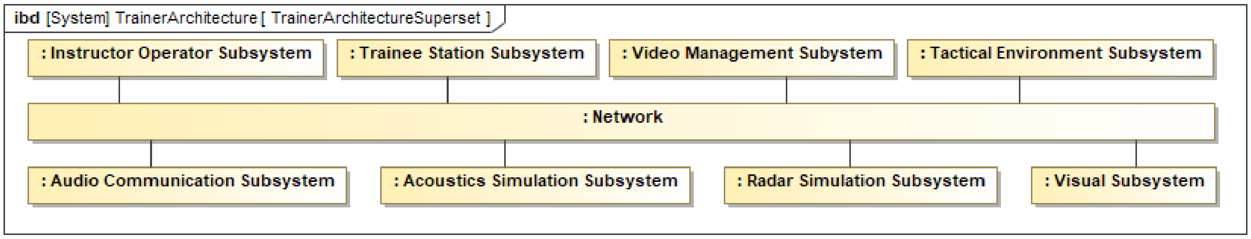

ibd [System] TrainerArchitecture [TrainerArchitectureSuperset]
: Instructor Operator Subsystem
: Trainee Station Subsystem
: Video Management Subystem
: Tactical Environment Subsystem
: Network
: Audio Communication Subsystem
: Acoustics Simulation Subsystem
: Radar Simulation Subsystem
: Visual Subsystem

Figure 3 - Operator Training Devices Systems Architecture

Figure 4 shows the four device types a student uses for operator training. First, the student begins in an Electronic Classroom (EC) using interactive courseware and instructor led training at individual workstations. Next, the student begins familiarization with the P-8 Operator Machine Interface (OMI) using the same EC hardware on a device called the Mission Systems Desktop Trainer (MSDT). Upon completion of MSDT training, the students begin using the Part Task Trainer (PTT) composed of an Instructor Operator Station (IOS) and three Mission Crew Workstations (MCWs). The highest fidelity trainer is the Weapons Tactics Trainer (WTT) containing five MCWs and three IOS's operating in a joint training scenario. Additionally, a portable version of the WTT called the Deploy Mission Rehearsal Trainer (DMRT) using scaled down and collapsible versions of the training hardware exists to allow training worldwide.

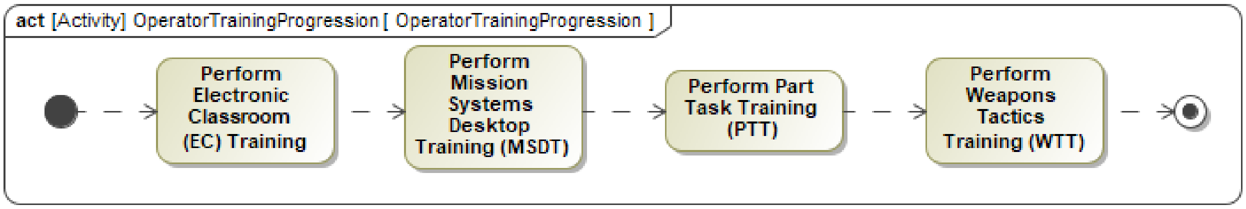

act [Activity] OperatorTrainingProgression [OperatorTrainingProgression]
Perform Electronic Classroom (EC) Training
Perform Mission Systems Desktop Training (MSDT)
Perform Part Task Training (PTT)
Perform Weapons Tactics Training (WTT)

Figure 4 - Operator Training Device Progression

P-8 Training Systems began their DE journey in 2017 by piloting the capabilities and value of using MBSE for the development of the DMRT. The DMRT was a new training device type that required a number of new technical challenges while trying to maintain training capability on a much smaller hardware footprint. Upon successful completion of the pilot program, Boeing decided to expand their MBSE efforts to modeling their existing WTT and PTT systems as baselines for future designs. As the number of device types and customers increased, clone and own resulted in significant duplication of design content between the models and growth in the variation between products in the product line. This clone and own approach provided initial reuse, but came with associated maintenance difficulties of keeping the content in sync. As mentioned above the three pillars of MBSE are language, tool and method. As P-8 transitioned out of the pilot phase they chose to use SysML, Cameo Enterprise Architecture and MagicGrid Framework as their MBSE pillars respectively. While the solution that follows will use these decisions, the strategy is agnostic to these decisions.

In a parallel effort in 2018, P-8 began taking an incremental approach to applying PLE starting with their Vehicle Simulation Software (VSS) subsystem. The team was focused on driving down the cost of an upgrade to the training systems for their customers using newer, more modern strategies. The team recognized the commonality between the products being developed for different customers and how a shared software baseline specifically configured for each customer could reduce the software design, implementation, testing and maintenance costs. The initial PLE solution was limited to the software implementation phase for the VSS subsystem. This is a good start, but misses out on other phases of the VSS's development lifecycle and the variation in the rest of the of the product line. In 2020, P-8 expanded their PLE implementation to their document-based Acceptance Test Procedures (ATPs). They were able to reuse many of the existing features to handle the variation in their tests. Since the PLE feature models had been limited to the software of a specific subsystem, additional effort was required to define features around the other variation in the test procedures. In early 2022, the team began exploring how to further expand their PLE solution to their MBSE models and their document-based cold start procedures.

As P-8's DE journey matures, getting more reuse of the design is becoming a focus area. The team understands the need to use PLE to manage the variation between the various device types and customers also called configurations. As the team investigates how to get better reuse from their design elements through PLE, they are also running into problems due to the growth in model size and need a solution to help them scale their MBSE modeling approach.

MODEL SCALABILITY

A key goal of effective organizations is to develop systems that meet the needs of their customers while finding cost and schedule efficiencies through efficient reuse of engineering artifacts. One of the best ways to achieve reuse of a complex system involves creating modular models by decomposing the design into smaller and smaller reusable elements. The Model of Models (MOM) methodology (Jackson et al., 2020) provides a number of reasons why monolithic models are problematic and discusses how elements from lower layer models are defined once and then reused by higher layer models. Within the MOM methodology Block Definition Diagrams (BDDs) are used to create compositions of reusable logical and physical blocks to create multiple combinations of integrated design elements.

Applying the MOM methodology to training devices includes creating the building blocks such as computers, displays, peripherals, etc. once and then reusing the elements to create larger reusable elements. The MOM methodology would then allow an IOS desk to be defined once as shown in Figure 5 and then reused within a training device and across multiple training devices.

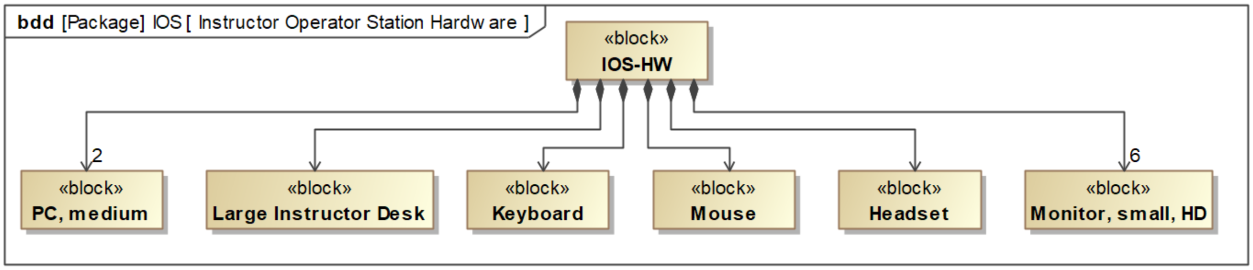


Figure 5 - Instructor Operator Station Hardware

The approach just described is a good start to improving reuse, but has one serious limitation in that the reuse is limited to the case where everything is exactly the same. A model contains additional design data for these elements such as how they are connected, how they trace to requirements, what software elements are allocated to the hardware elements, etc. When considering all of these additional modeling efforts, the cost of variation grows quickly. In order to achieve maximum reuse across system configurations and customers, additional techniques are needed that cannot be accomplished via modularity alone.

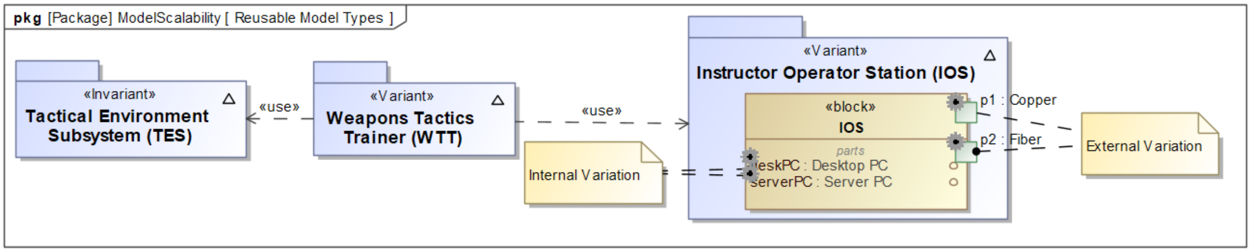
Product Line Engineering provides the solution to efficiently manage the variation challenges just described. Instead of duplicating designs for each configuration, a PLE factory approach models the superset design and generates customer specific designs from the superset model. Young and Clements described the details of combining MBSE and PLE as a simple and effective solution to achieving additional reuse. (Young & Clements, 2017). Their approach provides an excellent overview on using feature-based variation points to handle the different configurations. Their solution does not cover how to scale as models become larger and more complex resulting in the need to be split for modularity or reusability. Today's modern training devices need a combination of PLE to manage variation and multiple models to handle scalability and larger scale reuse.

COMBINED SOLUTION – PRODUCT LINE MODULAR MODELS (PLMM)

The Product Line Modular Models (PLMM) strategy is a scalable solution that enables maximum reuse by both breaking down models into smaller reusable parts and implementing PLE both within the reusable parts and as the parts are integrated together. Specifically, the PLMM strategy is comprised of two main types of reusable models, variant and invariant. An *invariant* model is a model that defines elements without the need to use PLE to manage variation within the model due to the standardization or simplicity of the content. Within a training product line, invariant library models of reusable simulation messages standards such as Distributed Interactive Simulation (DIS), High Level Architecture (HLA) and Computer Image Generator Interface (CIGI) are defined once and reused by any model. A *variant* model is a reusable model that requires PLE to manage the variation within the model using feature-based variation points. For example, the hardware for a student station often varies based upon the level of fidelity required such as between P-8's MSDT and WTT training devices.

A few additional concepts will be used to describe this strategy. First, the models are related to each other with a special type of dependency called a *usage* relationship. This Unified Modeling Language (UML) relationship indicates that one model, the client, requires another model, the supplier, to complete its definition. The model on the client end of the relationship is a *using* model and the model on the supplier end is a *used* model. From a modeling perspective, this means that elements in the *used* model are required to implement the design in the *using* model. The concept of a *used* role and *using* role will be continued throughout this PLMM strategy as the role of the model impacts how variation is managed. The second concept is stereotype, which is a mechanism to extend a model element's base implementation with additional details for the specific domain being modeled. The stereotype is represented visually by surrounding the stereotype name with double chevrons, for example <<stereotype name>>. Within the PLMM strategy, two primary stereotypes are used for the model to indicate variation, <<Variant>> and <<Invariant>>. The last concept is the use of feature-based variation points to indicate variation within a specific model element. Variation points allow the element to be either included or excluded from the specific configuration or allow a limited set of properties of the element to be changed, such as multiplicity. A model's variation is either internal, part of the white box design, or external, part of the black box design. From the using model's perspective, external variation is on the boundary of the used model and impacts at least one other design element. Internal variation is not as clear cut and requires addition analysis to determine impact to other models.

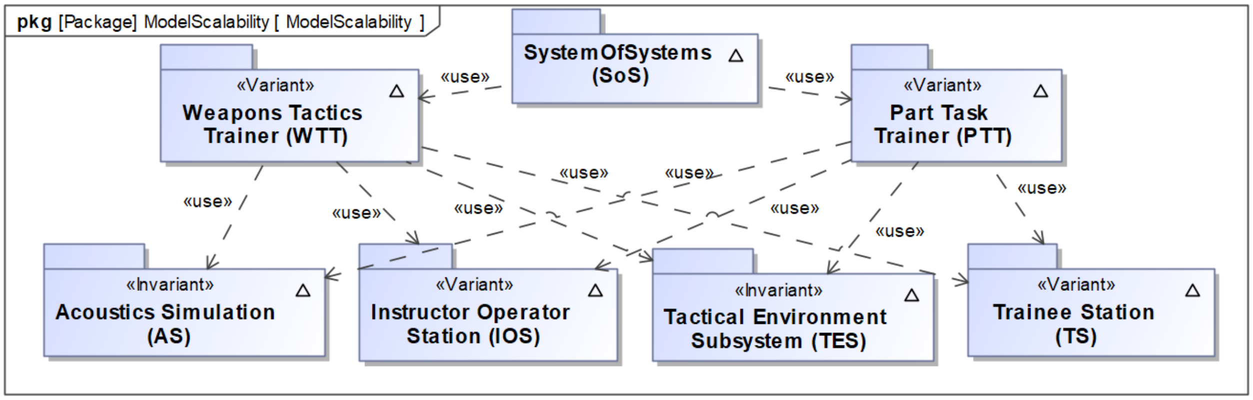
Figure 6 shows the usage relationship from the WTT model to the IOS and the Tactical Environment Subsystem (TES) models. The WTT is a using model and the TES and IOS are used models. This model usages implies that a WTT cannot be fully defined without content from the IOS and TES models. Additionally, the stereotype symbology is shown on each of the three models to indicate whether or not each model contains variation. In this example, the TES model represents a non-configurable COTS product so it uses the <<Invariant>> stereotype. Alternatively, the IOS is uniquely designed for the WTT trainer type so the IOS model uses the <<Variant>> stereotype. The TES model has no variation thus it contains no variation point symbology. The IOS on the other hand is a variant model and contains variation point symbology, a gears icon, on its proxy ports and parts. Since the proxy ports are on the boundary of the IOS subsystem, or part of the black box design, they are considered external variation. These ports vary based upon the required interface type of copper vs. fiber ethernet due to transmission speed requirements. This external variation impacts other design within the using model to ensure complimentary proxy ports for the other end of the port's connection. Alternatively, the part property variation points are not on the boundary of the model and represent internal variation of the IOS. For example, within one IOS configuration the desktop PC might be the appropriate solution, but for another configuration the PC needs to be in a server rack. Since this variation is internal, additional analysis is required to determine if this variation impact other models.



pkg [Package] ModelScalability [Reusable Model Types]
«Invariant»
Tactical Environment Subsystem (TES)
«use»
«Variant»
Weapons Tactics Trainer (WTT)
«use»
«Variant»
Instructor Operator Station (IOS)
«block»
IOS
parts
deskPC : Desktop PC
serverPC : Server PC
p1 : Copper
p2 : Fiber
Internal Variation
External Variation

Figure 6 - Reusable Model Types

As this solution scales, complex training devices must be broken down using a hierarchical decomposition into smaller models across multiple layers before modular reusable components are identified. Additionally, training devices are often integrated together into a system of systems (SoS) to enable joint mission training at a training facility. Many modern platforms scale to another layer to allow distributed training between training sites or between virtual and live players in a Live, Virtual and Constructive (LVC) configuration. Figure 7 shows a sample representation of the hierarchical modeling layers of SoS, systems and subsystems in which the SoS model uses system models that use subsystem models. This approach scales both up and down recursively across as many layers as needed. Additionally, a model can be both a using and used model based on the context of the relationship. In this example the system models are in the using role in the context of the subsystems, but in the used role in the context of the SoS.



pkg [Package] ModelScalability [ModelScalability]
SystemOfSystems (SoS)
«Variant»
Weapons Tactics Trainer (WTT)
«Variant»
Part Task Trainer (PTT)
«use»
«use»
«use»
«use»
«use»
«use»
«use»
«use»
«use»
«use»
«Invariant»
Acoustics Simulation (AS)
«Variant»
Instructor Operator Station (IOS)
«Invariant»
Tactical Environment Subsystem (TES)
«Variant»
Trainee Station (TS)

Figure 7 - Hierarchical Model Decomposition

The distinction between the two types of variation within variant models results in differences in how the features are organized in the feature catalogue. Like the models, the system features are hierarchical in that some features apply across the system and other are unique to a specific subsystem or component. Within the PLMM strategy, hierarchical feature models compliment the hierarchical MBSE models and multiple features models may apply to a single MBSE model. Features are evaluated based upon the scope of their impact as either impacting a single subsystem or multiple subsystems and then placed within the feature model hierarchy at the smallest scope possible. Features are organized using either a single definition of features (SDoF) or a multiple definitions of features (MDoF) approach. In the SDoF approach a feature is only defined once within the hierarchy so that all necessary design elements have access. The MDoF approach defines a feature multiple places in the hierarchy using constraint logic to ensure consistency. There are pros and cons of each approach so a tradeoff analysis must be performed based on the use case. Within the SDoF approach there is always an ASoT for the feature's definition, but the design element is now coupled to hierarchical feature models with a higher scope than the design element. This coupling makes reuse for another product line more challenging as a subsystem would also need system features. Conversely, the MDoF approach duplicates the feature so that each design element has access without dependencies outside the design elements scope, but requires additional logic to ensure all feature instances within a product are consistent. The MDoF approach makes more sense for products used on multiple product lines due the ability to define a complete set of complimentary features and MBSE models. Finally, the decision for MDoF and SDoF can vary throughout the hierarchical feature models.

In Figure 8 there are two feature models, one for the IOS network configuration using external variation and one for IOS hardware specific features using internal variation. Using the SDoF approach, the network configuration feature model would be defined once at the system level and all subsystems would inherit the decision. With the MDoF approach each impacted subsystem would own a copy of the feature model with constraint logic at the system level to ensure feature consistency. The IOS hardware design is reused outside of the P-8 product line so it makes sense to use the MDoF approach for this feature model. Other P-8 specific subsystems, such as the trainee station, are not expected to be reused outside the product line, so from the trainee station's perspective, the SDoF approach makes more sense. When making the decision of feature definition approaches, multiple perspectives often need to be considered. Since the IOS Hardware feature model represents internal variation additional analysis must be performed.

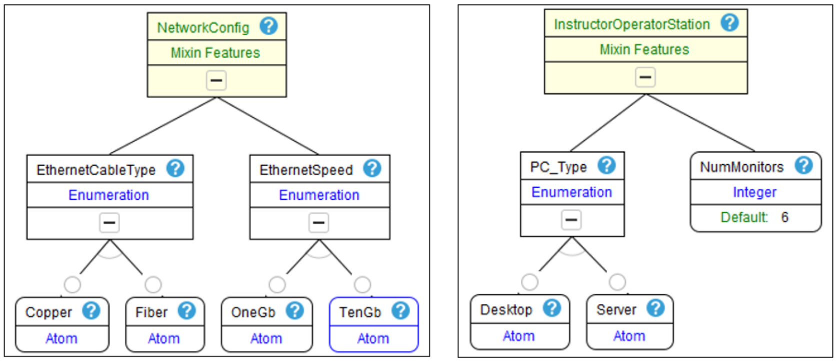


Figure 8 - Example Feature Models

Figure 9 shows how using the PLMM strategy a system level behavior is modeled with two different TES subsystems. System level variation allows either TES A or TES B to perform the role while the rest of the design remains the same. While the TES model is an invariant subsystem model, it becomes a variant at the system level. Nested swimlanes have been used for illustrative purposes to show both the responsible subsystem in the inner, or bottom, swimlane and the owning model for the subsystem in the outer, or top, swimlane. The activities elaborating each of the call behavior actions for the subsystem would be contained within their subsystems model with complementary variation management. For example, the "Detect Surface Contacts" activity within the Trainee Station Model has variation to handle either "Surface Entity Data A" or "Surface Entity Data B" as inputs. Since the decision on which tactical environment to use is a system level decision and the Trainee Station Subsystem is unique to P-8, the SDoF approach is used. Without PLE, all the forks and joins, object and control flows, traceability to other design elements, etc. would need to be duplicated and kept in sync between duplicate system level activities. Finally, this modular approach enables TES A or TES B to be easily reused on another product line along with all of its internal design details.

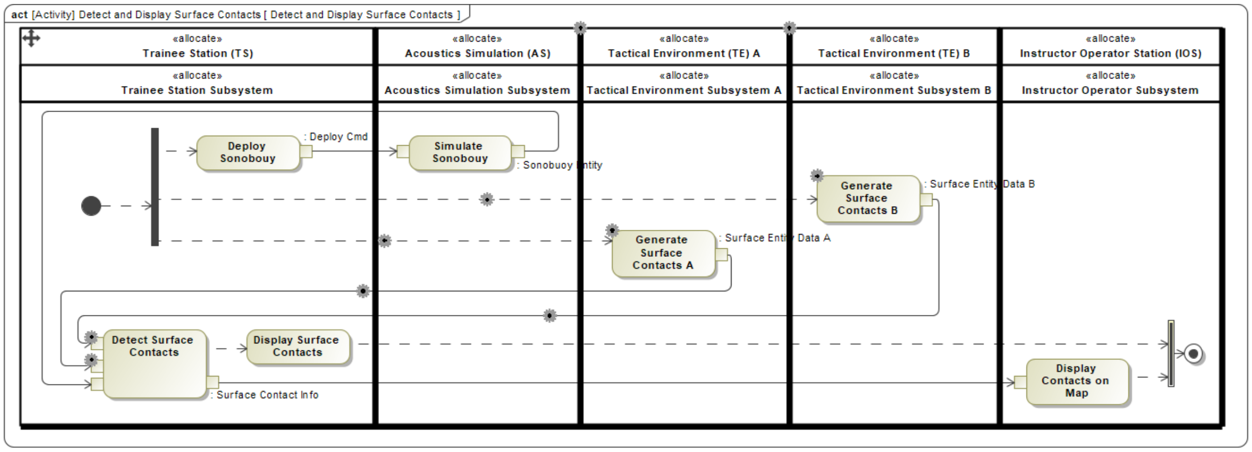

act [Activity] Detect and Display Surface Contacts [Detect and Display Surface Contacts]
«allocate»
Trainee Station (TS)
«allocate»
Acoustics Simulation (AS)
«allocate»
Tactical Environment (TE) A
«allocate»
Tactical Environment (TE) B
«allocate»
Instructor Operator Station (IOS)
«allocate»
Trainee Station Subsystem
«allocate»
Acoustics Simulation Subsystem
«allocate»
Tactical Environment Subsystem A
«allocate»
Tactical Environment Subsystem B
«allocate»
Instructor Operator Subsystem
Deploy Sonobouy
: Deploy Cmd
Simulate Sonobouy
: Sonobuoy Entity
Generate Surface Contacts B
: Surface Entity Data B
Generate Surface Contacts A
: Surface Entity Data A
Detect Surface Contacts
Display Surface Contacts
: Surface Contact Info
Display Contacts on Map

Figure 9 - Reusable Behavior Example

The PLMM Strategy applies well to the P-8 due to the large amount of reuse of hardware and software across the training devices. Using this strategy, models are designed to be as complete and reusable as possible. Each model contains the four main pillars of SysML: structure, behavior, requirements and parametrics. Additionally, the model encapsulates the SysML traceability relationships, such as refine, satisfy and allocate. This complete solution allows model reuse with minimal effort to connect external interfaces and perform traceability between the models.

BENEFITS AND LESSONS LEARNED

The use of the PLMM Strategy provides an opportunity for product lines to reduce the overall size of the models providing the team more time to focus on the technical work and innovation needed to advance the product line. As the P-8 program continues its digital journey, innovation is leading to benefits and lessons learned including increased consistency, efficient reuse, reduced configuration complexity and improved collaboration.

All DoD contracts require a set of Contract Data Requirements List (CDRLs) to be delivered as part of the development effort, such as interface designs and system designs. The information for these CDRLs is often pulled from a variety of sources and put into a document for the customer while requiring manual effort to make sure the information is consistently documented. The MBSE models are inherently consistent by nature of the rigor MBSE imposes and contain most, if not all, of what is needed to deliver these CDRLs. The best option would be to deliver the model in place of the documents, but for contracts requiring paper documentation this isn't possible. P-8 is developing innovative solutions to automatically generate documents from the model for System/Subsystem Design Documents (SSDDs), Interface Control Documents (ICDs) and Interface Definition Documents (IDDs) in accordance with MIL-STD-498. Generating these CDRLs from the model allows the content to be updated once and then all impacted documents are updated automatically within a matter of minutes. For example, the IOS's network interfaces appear in the ICD for the IOS and the SSDDs for the PTT and WTT. With the PLMM approach, this is captured once in the IOS model and then the IOS model is used by the PTT and WTT models to generate the three CDRLs each. Furthermore, by applying PLE to the models, the CDRLs automatically become tailored to the individual customers through the inclusion of only the appropriate features for the specific product. When the IOS needs an additional network interface for a new feature, the interface is added in the model once and then through PLE the same three documents are generated with or without the interface based upon the inclusion of the feature. In this simple example, six documents would previously have been manually updated each time something changed, but now the change is made once. These example changes have to be made regardless of the CDRL delivery since they are part of the actual design of the product, so by eliminating independent work activity for the CDRL reduces duplicate efforts.

In order for P-8 to provide the varying levels of fidelity needed across the trainers, hardware variation is used to provide solutions with smaller footprints and lower costs. Since the training objectives are very similar or even the same, the software is reused to provide consistent functionality for the student. Using PLMM, this was accomplished by keeping hardware and software designs loosely coupled and in separate models. These designs are integrated as used models in a using model by allocating the software design to the hardware design. This approach allows for better modularity and reuse, only requiring modifications to the allocation relationships from the software to the hardware. For example, a common practice today is to move more and more training from dedicated hardware into cloud-based assets. The separate software model is a used model in both configurations, but remains unchanged. Similarly, the same software can be reused and allocated to hardware configurations for a desktop trainer and a full weapons training device. The same concept applies when the hardware is the same and the software is different. For example, the hardware elements are modeled once and different versions on TES software are allocated in different configurations. Also, this solution is very useful in the case where a classified model for the software replaces an unclassified version that adheres to the same interfaces. Much of the design can be developed and reviewed in an unclassified environment and then only the classified portions of the model need to be developed and maintained in the classified environment.

The P-8 VSS team has been using PLE for over four years to manage over a million lines of code across multiple software repositories. The biggest savings have come from the reduction in the number of parallel branches and the complexity that come with keeping the duplicate code in sync across multiple customer branches. Using PLE, a single development branch eliminates mundane merge activities between branches and duplication of coding. Additionally, errors can be fixed once for all customers resulting in faster cycle times. Recently P-8 bid an existing feature for a new customer and provided a reduced VSS coding estimate for that feature because of PLE. P-8's PLE implementation on more than 100 Acceptance Test Procedures (ATPs) has seen similar benefits in reducing mundane activities and faster cycle times. P-8's digital engineering journey is still early, but applying the PLMM strategy to just to the operator training devices results in reducing the need for multiple system models, one for each variant, to just a single model. Within P-8 there are more subsystem models than software modules, so the complexity to manage these usages would exceed that of the software suggesting the MBSE benefits are at least that of the software.

P-8 is one of many training platforms within Boeing and collaborating across the organizations and sharing models is generating additional reuse. Through the PLMM strategy, the hardware engineering team defines a single model to represent the logical hardware elements used across training platforms. The platform teams include this model and use the elements for their design. For example, the hardware team has defined a set of standard PC configurations and the P-8 engineers are designing their system using these shared element definitions. Additionally, these elements contain objective descriptions to avoid ambiguity in the interpretation between the teams. For example, CPU speed is defined in the model with values of low, medium and high with each specifying the range in GHz for the processors. This strategy is increasing the collaboration between the teams and driving consistency in the hardware design. As hardware configurations become obsolete, the hardware team can update the shared model once for all platforms.

Finally, the PLMM strategy applies to more than just the standard elements being designed such as the various systems, subsystems, components, etc. Variant library models should be created for custom message interfaces which are reused across product lines but contain variation from product to product. For example, the DIS standards allow for custom defined variable records to be added to many of the standard messages. These can be modeled once for all product lines with variation points used to manage the differences in message structure. Furthermore, modeling requires processes and standards to ensure consistency within the model. While there are SysML and UML standards, there are many aspects of the modeling that remain flexible and the flexibility will limit the reuse of the model. Processes and standards can be modeled once and then reused across products as variant models that allow the customizations often needed. These additional uses of models require a core team to manage these models that are looking at the models from an organizational perspective and not focused on the individual platforms.

SUMMARY

The large-scale P-8 training system is transforming system development through Digital Engineering that includes Feature-based PLE and MBSE using an approach called PLMM. Adopting this approach has already led to many benefits including reduced complexity, cost reductions, improved team coordination and innovations with automation and opportunities for synergy with other Boeing training system product lines. The P-8 training system had demonstrated it is well-suited for PLE with the many sources of variation that include platform derivatives, simulation fidelity, deployment environments, export control, security and many subsystems. Developing digital models of the system along with digital models of variation are unlocking new possibilities and innovations that were previously unavailable. Future system variants can be delivered at lower cost with reduced schedule risk.

We believe future training systems for other platforms will see similar results using PLMM. However, those future programs would benefit from requirements stipulating use of Digital Engineering techniques such as PLE and MBSE. A solid Digital Engineering starting point will promote healthier reuse and sharing that ultimately leads to reduced sustainment costs and lower complexity that will unlock new innovations for the product line.

FUTURE WORK

Future efforts include additional guidance for performing impact analysis on internal variation, best practices for sharing models between training devices and aircraft programs, how the modular models align to MOSA systems and how the PLMM strategy evolves throughout the product development lifecycle including application to digital twins.

ACKNOWLEDGEMENTS

We would like to thank Dr. Paul Clements for his career focused on education, advocacy and excellence in Software Architecture and Product Line Engineering. We would like to thank the following reviewers: Thomas Gasque, Nicole Lee, Rebecca McBride, Peter Novak and Brad Smull.

REFERENCES

BigLever Software. (2022). *BigLever Software Gears* (version 8.5.2). http://www.biglever.com/solution/product.html.

Clements, P., (2015). *Product Line Engineering Comes to the Industrial Mainstream*. 2015 INCOSE International Symposium, Seattle.

Clements, P., Gregg, S., Krueger, C., Lanman, J., Rivera, J., Scharadin, R., Shepherd, J., Winkler, A. (2014). Second Generation Product Line Engineering Takes Hold in the DoD, *Crosstalk, The Journal of Defense Software Engineering,* vol. 27 (1).

Clements, P., Krueger, C., Shepherd, J., Winkler, A., (2013). *A PLE-Based Auditing Method for Protecting Restricted Content in Derived Products.* Proc. 2013 Software Product Line Conference, Tokyo.

Clements, P., and Linda Northrop, (2002). *Software Product Lines: Practices and Patterns*, Addison-Wesley.

Dassault Systèmes. (2022). Magic System of Systems Architect (version 2021x, previously known a Cameo). https://www.3ds.com/products-services/catia/products/no-magic/magicdraw/.

Delligatti, L. (2014). *SysML Distilled: A Brief Guide to the Systems Modeling Language*. Addison-Wesley Professional.

Gregg, S., McAfee, M., Teaff, J., Hartley, D., Pitz, R., Clements, P. (2020). *Patterns for Success in the Adoption and Execution of Feature-based Product Line Engineering: A Report from Practitioners*. 2020 INCOSE International Symposium, Cape Town.

INCOSE International Working Group for Product Line Engineering. (2019). *Feature-based Systems and Software Product Line Engineering: A Primer.* Retrieved from INCOSE-TP-2019-002-03-0404: https://incose.org.

INCOSE International Working Group for Product Line Engineering. (2007). *Systems Engineering Vision 2020*. https://sdincose.org/wp-content/uploads/2011/12/SEVision2020_20071003_v2_03.pdf,

International Organization for Standardization. (2021, April). ISO/IEC 26580:2021 Software and systems engineering -- Methods and tools for the feature-based approach to software and systems product line engineering. Retrieved from https://www.iso.org/.

Jackson, A., Palmer, B., Cobb, D., (2020). Model of Models Methodology: Reuse Your Architectural Data. *American Journal of Management*, vol 20(5).

Krueger, C. and Clements, P. (2013). *Systems and Software Product Line Engineering*. Encyclopedia of Software Engineering, Philip A. LaPlante ed., Taylor and Francis. www.biglever.com/wp-content/uploads/2019/01/Encyclopedia_Software_Engineering_P LE_Chapter.pdf.

Krueger, C. (2022). *How to use a Modern Digital Approach to System Family Engineering to Stay Ahead of the Competition.* BigLever. https://biglever.com/learn-more/biglever-videos-product-collateral/webinar-modern-digital-approach-to-system-family-engineering/.

Lanman, J., Kemper, B., Rivera, J., Krueger, C., (2011). *Employing the Second Generation Software Product-line for Live Training Transformation.* Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) 2011.

Linden, F., Schmid, K., Rommes, E. (2007). *Software Product Lines in Action*, Springer.

Office of the Deputy Assistant Secretary of Defense for Systems Engineering. (2018). *Digital Engineering Strategy*. https://ac.cto.mil/wp-content/uploads/2019/06/2018-Digital-Engineering-Strategy_Approved_PrintVersion.pdf.

Teaff, J., Young, B., Clements, P. (2019). *Applying Feature-Based Systems and Software Product Line Engineering in Unclassified and Classified Environments.* 2019 INCOSE International Symposium, Orlando.

SPLC Product Line Hall of Fame, https://splc.net/fame.html.

Young, B., Clements, P. (2017). *Model Based Engineering and Product Line Engineering: Combining Two Powerful Approaches at Raytheon*. 27th Annual INCOSE International Symposium.