Carnegie Mellon University
Software Engineering Institute

A Strategy for Component Product Lines: Report 2: Specification Modeling for Components in a Component Product Line

Sholom Cohen
John Hudak
John McGregor

May 2023

SPECIAL REPORT
CMU/SEI-2023-SR-008
DOI: 10.1184/R1/22780034

Software Solutions Division

http://www.sei.cmu.edu

Design: REV-03.18.2016.0 | Template: 03.22.2022

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

This report was prepared for the SEI Administrative Agent AFLCMC/AZS 5 Eglin Street Hanscom AFB, MA 01731-2100

DM23-0533

Table of Contents

List of Figures

List of Tables

Abstract

This report is the second in a series to establish a strategy for creating and managing a component product line (CPL); to define a modeling approach for specifying components; and to describe a governance method for making decisions to select a product line, determine the feature sets, and maintain the supplier–user relationship. The strategy considers who creates models, how they are represented, who uses them, and how they are evaluated for use. This report introduces the concept of a "model chain" to describe the models used in specifying a CPL. The model chain concept extends to show relationships between models, how they evolve, and the realization of architecture requirements through the creation–evolution process. The report applies the principles of this framework to three example governance scenarios based on a specification of a common industry standard. The examples include specification views to support the model understanding that a CPL specifier must provide and to create feature, function, and other model analysis attributes so that the supplier and potential users can assess whether or how well a CPL satisfies its specification. The analysis results become key factors in the decision-making process of component selection for integration into a weapon system.

1 Introduction and Motivation

A component product line (CPL) consists of a set of components that satisfy desired capabilities and constraints, called a feature set, for a range of systems. Those systems constitute the potential users of component instances from the product line. Strategic reuse consists of successful integration of CPL instances within a specific user system to satisfy the requirements of that system.

This view of strategic reuse assumes a context of component use with the following characteristics:

- Systems that will use the components will not always be known to the developers of those components. Components must be specified, developed, and supplied without specific knowledge of the potential user systems. Where acquirers or integrators perform component development tasks, they should avoid applying system-specific knowledge that couples the component to a system.
- Acquirers of systems that are potential users of components must be able to evaluate and select components for their systems. The selection must be made through the component specification and through supplier specification and design models. The specification models define the following: capabilities supplied by components within a CPL, the feature sets covered by the components, knowledge of planned variations within a feature set supported by the components, and externally visible properties of components within a CPL. Supplier design models include design and implementation information that can be used to evaluate and select from among available components.
- Integrators develop systems to the system requirements through use of system-unique components and use of components from CPLs. They may use integration-ready components—those already implemented and packaged for system use—or components from a product line production capability that can be configured and instantiated as integration ready. In addition, an integrator or other developer may make system-unique extensions to a CPL component for specific system use.

Can a model-based approach guide the development of product lines of components? A complete approach must cover CPL specification to support acquisition decisions. Acquirers make those decisions before or in parallel with the use of specification models by suppliers and integrators of components. The modeling approach must also apply to the community involved with designing, implementing, and sustaining product lines of components used by the government.

The approach described in this report on product line modeling concentrates on the Component Product Line Specification Model (CPLSM) and use of that model. Example users of the CPLSM include acquirers who specify, evaluate, and select appropriate components; component suppliers who determine suitability of existing components to satisfy a specification; suppliers who wish to build a production capability for a full product line of components; and weapon system (WS) integrators. This modeling approach also includes guidance for component design during the implementation process.

The component specification modeling approach helps achieve desired results of the CPL strategy established in Report 1 of this series: *A Strategy for Component Product Lines: Report 1: Scoping, Objectives, and Rationale* [Cohen 2022]. That report, called the "Strategy Report" throughout, outlines the overall intent and game plan for implementing a CPL approach (the "what" and the "why"). The component product line strategy is to apply components from a CPL. Each component is derived from a CPLSM to support modularity in system development and open systems that can be easily modified and upgraded.

This report, called the "Modeling Report," describes a modeling approach that also aligns with the report *Component Product Line Governance* [Schenker 2023], referred to throughout as "Governance Report." The Governance Report provides critical insight into the process of government implementation for a successful product line strategy. It is the "how to" and includes a series of governance CPL scenarios.

The Modeling Report supplies guidance to the specification modeling teams, suppliers, acquirers, and integrators for developing models for CPLs. Scenarios in this report address each of the governance scenarios to account for core requirement specification, variation modeling, and feature set creation by the CPL specification modeling team. The scenarios also cover handoff of specification models to WS acquirers to support acquisition decisions, to suppliers for component instantiation and implementation, and to integrators who may serve as suppliers of components as well as WS integrators, applying supplier-developed components and their own component developments to perform integration.

This report establishes a modeling approach for the component model chain. The model chain includes the CPLSM model contents. This report uses the contents described in the Model-Based Product Line Engineering (MBPLE) approach captured in Cameo profiles [MBPLE 2021], but other modeling approaches may also be applied, such as the ISO reference model for product line engineering and management [ISO/IEC 2015]. MBPLE uses the following basic concepts:

- The features model contains "what's available for choosing" for configuring a component. A collection of related features may be categorized as feature groups [MBPLE 2021].
- Feature sets are actual feature choices, recorded as configurations for specific components.
- The multi-variant 150% component model is a system model annotated with variation points (VPs). VPs identify a part of the CPL 150% model that is variable. If the model element—such as a requirement, function, or connection—exists only in some instances of a CPL, the modeler will demarcate the CPL element with a VP.

The CPL specification is used by the specification team, acquirer, supplier, or integrator to create the variant realization or instance model with the model chain. This report provides development approaches for each model in this chain for the CPLSM. The component instance model is applied through design and implementation practices, which are not covered in this report, to create the integration-ready component.

The goal for this report is to establish a CPLSM development and integration approach, one that maintains an up-to-date authoritative source of truth for the CPL at each development step. Any additions, corrections, or derivations cannot become part of a revised CPLSM unless and until the developer traces those revisions back to an authoritative source. This trace back to source information for a CPLSM goes not only for models of requirements. All models form a chain of information content. Design models, the integration-ready component, and the implemented component derived from the CPLSM must link their content back to models of requirements, design, or other sources.

The resulting model chain can be used as a source for complete and correct verification and validation activities. CPL verification answers questions about the component specification models: Do they follow modeling guidance to address the requirements, feature set content, and expected behavior of components for specification users? CPL validation applies analysis, demonstration, inspection, simulation, and test to component instances to assure that components correctly address user needs: Are the components to be built or already built correct for the WS? Verification is performed by the specification modeling team. Validation is performed by suppliers against component specification requirements and by the acquirers or integrators, independently of the component supplier, against WS requirements.

The report addresses the key questions listed in Table 1 with respect to components in a CPL.

Table 1: Key Questions Addressed by This Report

• What specification content is needed to model the functional and nonfunctional requirements of the components? • Which models and submodels are needed to capture that content? • What questions should each model or submodel address? • What design content should be modeled? • How does asking the correct questions lead to a design that satisfies the specification and adequately informs potential users for component evaluation? • What are the characteristics to look for in a CPL model chain to support integrators of the component? • Is there a minimal submodel set to define the model chain?

The structure of the report is as follows:

- Section 1, Introduction and Motivation, explains the rationale for establishing a modeling approach for the component model chain.
- Section 2, Modeling Approach for CPL Specification, describes product line suppliers and users; roles and responsibilities; and variation and variability for use by the ecosystem of the CPL Organization, modelers, developers, and users.
- Section 3, Creation of Models in a Model Chain, establishes the approach to creating the model chain for the CPL, the scope and types of models, and the qualities those models should address.
- Section 4, Scenarios, illustrates the workflow of model development and model analysis by applying scenarios from the Governance Report and providing an example model chain based on the ARINC-615A Data Loader standard [ARINC 2007].
- Section 5, Summary, concludes the report and lays out the next steps for this research.
- Appendices offer supplementary information, including a use case scenario description, acronyms, and glossary.

2 Modeling Approach for CPL Specification

This section of the Modeling Report describes the CPL supplier and user sides, roles and responsibilities for modeling, and the need to model for variation across the CPL. It provides an approach that applies objectives contained in the Strategy Report in the remainder of this Modeling Report. The Strategy Report includes technical and organizational elements for an approach guided by models for component specification.

2.1 The CPL Model Supplier and User Sides

The approach establishes two sides of the CPL strategy: the supplier side and the user side. On the supplier side, a government CPL Organization consisting of a product line champion, product line manager, and CPL specification teams creates CPLSMs. These roles are defined in the Strategy Report and refined in the Governance Report. The supplier side also includes suppliers who use CPLSMs to create designs and integration-ready components. The user side includes the acquirers and integrators of WSs who apply the models for WS specification, component design, and implementation and integrate the integration-ready components for WS development and implementation.

The CPL specification modeling team defines functionality and variation across the CPL to create the CPLSM. This definition includes the development of features, features sets, and the 150% model identified in Section 1: Introduction and Motivation and detailed in Section 2.3: Establishing Features and Variations for the CPL. The supplier side is guided by the following principles:

- Model-based systems engineering and other related practices refine the CPLSMs into instance models for the CPL as needed for use in a specific WS.
- Suppliers propose existing components as instances of the CPL, develop new components to the CPL specification, or create a product line capability for delivering multiple component instances for the CPL.
- Specifiers and suppliers of components do not know in what systems the components may be used, and details of the systems in which the component will be integrated are not known. Variation in such use must be identified and modeled as part of the model chain.
- Component designs are part of the model chain that applies the component instance model and leads to integration-ready components. Providing these designs for review gives the government an opportunity to weigh in on the design approach before going to code or fabrication.

On the user side, WS acquirers, suppliers, and WS integrators will only know

- the feature sets provided by the CPL
- externally visible properties of a component (interface and behavior attributes)
- mechanisms to use those properties to tailor a component to a specific use based on published variation properties of the component
- the infrastructure and mechanisms into which the component must be placed so that it can operate correctly and so that the component can be extended, if needed

- architecture patterns, execution platforms, and related software artifacts, such as libraries, in which a component interoperates

These users apply the CPLSM, CPL designs, or an integration-ready component. When a user needs component features that are incomplete, the user may request updates to the component. These updates may be addressed in a variety of ways:

- The CPL manager and CPL specification modeling team add VPs and features to the CPLSM.
- The suppliers address new feature sets.
- The integrator adds features outside the component where the component model and instance offer interface or adds other connections to support this integrator role.

An important consideration in development and use of the CPLSM is the ability to handle change. The need for change is a primary driver addressed in Section 4: Scenarios for several reasons:

- Change occurs as a system evolves from specification to actual implementation.
- Change results from requirements of new WSs that were not part of the scoping process but are following the CPL approach. (A new WS may have requirements that drive new features at a VP or other changes to the specification model.)
- Change occurs once those systems are in operation.

Satisfying specific architecture requirements for modification and extension provides support for those changes, accommodating the specification change that will occur during the lifecycle of development and operation. Modeling specification and design views enable change, whether planned or unplanned, that is not bound to an implementation. For product line component specification and use, the approach must accommodate planned change with alternative components that support a common core of requirements and must specify known variation across the common core as well as optional features.

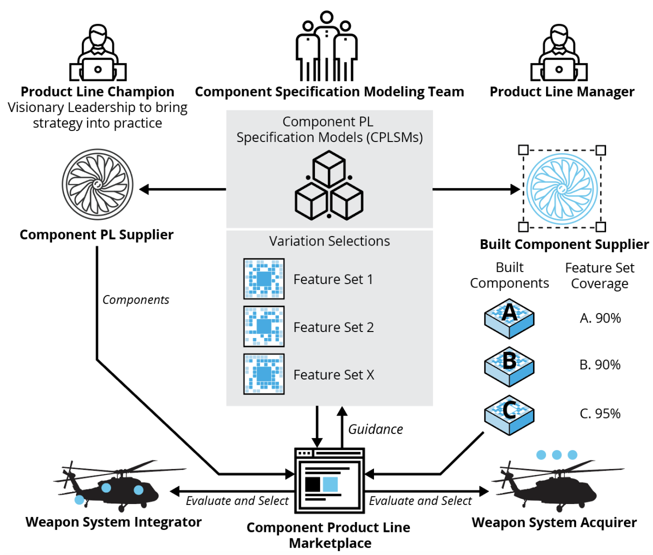
Integrators for the system using a component can accommodate planned change by selecting alternative variations within the component. If these are not satisfactory, the integrator may choose to integrate an instance component from the same product line that offers a different feature set, one that can accommodate the required change to the system. Unplanned change must satisfy component requirements for modifiability and extensibility. This unplanned change may drive changes to the CPL specification. Where the CPL suppliers have applied appropriate architecture approaches, the existing components may accommodate the change, resulting in alternative component instances. In extreme cases, changes may require a new product line.

2.2 Roles and Responsibilities in Modeling for CPLs

The CPL modeling approach includes model specification and development activities. In support of the Strategy Report, the modeling approach presented here establishes a basis for strategic reuse of software CPLs. Figure 1: Product Line Strategy: From CPLSM to Integration-Ready Components in the Marketplace illustrates the chain of models that lead from scoping decisions about the CPL to models that support WS acquisition and integration. The figure represents the major roles in the strategy as the components progress from concepts to concrete models:

- The product line champion and managers identify and scope CPLs. (See Section 4: Scenarios.)

- The CPL specification modeling team creates and delivers specifications and features sets. (See Sections 2.3: Establishing Features and Variations for the CPL and 4: Scenarios.)
- Component suppliers derive design models of structure, behavior, and other views from the specifications and feature sets. (Supplier design modeling is not covered in this report.)
- WS acquirers and integrators apply the component specifications and supplier models for evaluating and selecting implemented components that are ready for specialization and integration. Specialization and integration by integrators may include applying adapters but may also require developing and implementing communication and connectivity capabilities from the system to components to enable component integration and use. (See Section 4.5: Scenario 3: Application of Component Product Line to a Weapon System.)



The scoping decisions for CPLs are made by the product line champion and manager (Sections 3.1, 4.1).

CPLSM teams create the specification with variations and feature sets to be captured by specific components (Sections 2.3, 3, 4).

Component suppliers derive components or a component production capability from the CPLSM and apply the model chain concept for inclusion in the CPL marketplace (Sections 2.3, 4.5).

The integrator and acquirer apply the component specification model and marketplace models to evaluate and select CPL components for the weapon system (Sections 2.3, 4).

Figure 1: Product Line Strategy: From CPLSM to Integration-Ready Components in the Marketplace (Derived from Strategy Report Figure 1 [Cohen 2022]. PL = product line.)

Section numbers in Figure 1 reference the portions of this report that describe modeling approaches for these roles. The two categories of suppliers are a key consideration for the product line:

1. The built component supplier (right side of figure) offers components that previously existed and have been modified or enhanced to address a specific feature set established by the specification team.
2. The component PL supplier (left side of figure) offers to create new components based on a product line production capability that can produce components that address specific features sets on demand.

Identifying product lines and defining capabilities entails significant up-front analysis to arrive at the appropriate component scope. Scoping defines capabilities provided by a potential product line, determines relationships between and among those product lines, and specifies interactions of potential components with the WSs that use those components. These interactions include evaluation of existing products, surveys of WSs that may have need for the components, and analysis of legacy product specification. Information captured during these interactions serves an input to the CPL specification modeling team.

Development of CPLSMs applies analysis to document and model various component requirements: function, behavior, features, variations, and so on. The analysis may also determine that one component may be integrated with other components, either to extend the component features or as a dependent component. The CPL champion, managers, users, and others from the ecosystem participate in the scoping actions and decisions. The All Viewpoint (AV-1, text) and Operational Viewpoint (OV-1, graphics) models from the Department of Defense Architecture Framework (DoDAF) may be applied to initiate definition and modeling of product line scope [DoDAF 2010].

Table 2: Stakeholder Roles lists key stakeholder roles and describes the information needs for each role. A stakeholder's needs reflect the role each plays with respect to the CPL models. (See Table 4 in the Strategy Report for details.) The product line component specification team applies the CPL scoping decisions and must create specifications of common as well as varying features for the CPL components. A feature is defined as a "prominent or distinctive user-visible aspect, quality, or characteristic of a software system or systems. ... Features are used to express product differences in all lifecycle phase artifacts. … Features are used to describe the products' variations" [BigLever 2022]. The variations that exist across the component in the CPL can be defined by features—those that are required of each component (mandatory or core), optional to a component, or alternative, where one or more features may be selected from a list. The complete set of features will not apply to any one component, but a selection of features—a feature set—will define a specific component as an instance model within the CPL. The CPL Organization, suppliers, and integrators perform a related set of activities, described in Section 2.3: Establishing Features and Variations for the CPL (Figures 2–5), to

- create the core required features in the CPL
- extend required features with options and alternatives
- select among the features to assign a feature set
- create the instance model and extend with WS specific features

Maturity and expertise of developers and other stakeholders will vary with respect to the domain(s) covered by the CPL or WS under development. The modeling for the CPLSM must consider these stakeholder capabilities. They will influence the scope of the CPLSM and other models needed by those stakeholders. Variation modeling must address the CPL stakeholders on both the supplier and user sides. A VP to capture deployment features affects the scope of a component instance. For example, deployment features can indicate direct network connection or indirect interaction through a network service, implying a transport mechanism to connect to a network. While some stakeholders need to see only the connectivity as a point of variation in a functional specification, component suppliers or WS integrators will expect to see sequence diagrams or other behavior models in the CPLSM. An expert in the behavior defined in the sequence will want to know message flows, timing, and exchanges. A message roll-up abstraction may be appropriate for a stakeholder who is concerned with only basic interactions. The expert, in this example, will expect to see modeling of the full protocol for network interoperation. The expert needs system context—information that can be added prior to solicitation or evaluation but is not necessary for the core features. This deployment variation is a focus of the exemplar presented in Section 4.3: Scenario 1B Example—The Data Loader CPL 150% Model.

Table 2: Stakeholder Roles

Needs and Responsibilities/ Role	Information Needs (Questions That Require Answers)	Responsibilities	Authority
Product Line Champion	feature commonality and variation analysis (across all domains)	Maintain scope and focus of all PLs.	Commit start-up funds.
Product Line Manager	anticipated product needs from WS acquirers	Develop component scoping. Assign and manage specification teams. Work with acquirers and integrators to support use of components from each PL.	Review models used to specify components and review components developed by suppliers. Reject proposed components that violate the scoping rules. Ensure that updates are propagated to all eligible products.
Component Specification Modeling Team	component scope vocabulary and standards-based concepts for the component domain	Create specifications using accepted vocabulary. Maintain integrity of the model chain.	Develop component specification.
Component Supplier	CPLSM 150% model or instance model	Select features from the CPLSM as “instance” requirements.	Design and implement a PL component or component capability. Recommend changes to specification.
WS Product Acquirer	1. During WS specification: CPLSM must support determination that PL component specification can be used for WS specification (with focus on functional boundary, features sets, interfaces, and quality attributes). 2. For WS design and implementation: Evaluate and select components or component capability from the marketplace for use by integrators.	Use CPLSMs to select features for component instances of a WS specification. Identify or acquire PL components that conform to the specification.	Accept or reject PL components proposed by integrators.
WS Integrator	Documentation of PL component or component capability from supplier is sufficient to determine feasibility of PL component integration and mechanisms to integrate a PL component into the WS, including component tailoring as supported by interfaces.	Deliver WS to the acquirer through integration of components from the marketplace along with components that are not in a marketplace PL.	Accept or reject PL components proposed by suppliers.

Note: PL = product line.

Modeling information captured in the CPLSM must address the needs of acquirers and integrators of systems that will potentially use the components. Acquirers of systems considering component use based on the specification will ask questions such as these:

- Do models address the needs of my system as a potential user of a component?

- How will component suppliers assure that the delivered instances do not differ in substantial ways from the original specification of that component?
- What approaches will the CPL Organization follow to maintain the component model description from specification to implementation?
- Will component instances support change driven by system change?
- Do feature sets of alternative components within the product line support that change?

A component being integration ready means that the WS can integrate the component even in cases where those systems are not known to the component specification or component supplier organizations. Acquirers or integrators of a WS evaluate and select components from CPLs based on feature sets offered by a component. The components must also satisfy architecture requirements that can answer the above questions. The WS integrators must have internal modeling approaches to integrate component models into their system models. A WS may not be fully modeled. However, the integrator should still be able to determine that the CPL component can be integrated and will support the acquirer's WS needs.

2.3 Establishing Features and Variations for the CPL

The CPLSM includes a feature model containing capabilities needed by components of the CPL. The roles specified in Table 2: Stakeholder Roles apply these variations in different ways:

- The modeling team for the CPLSM must address the existence of the variation and related features.
- The component supplier applies the variation to differentiate components in the supplier CPL.
- The acquirer uses the availability of the feature sets to guide the WS specification. The acquirer then determines availability of already existing integration-ready components that satisfy the selected feature set.
- The WS integrator collaborates with the acquirer to support component feature set selection and estimate the impact of that feature set on system attributes. The integrator also determines what design mechanisms will be needed to integrate the component selected by the acquirer.

The remainder of this section describes the different classifications of features: core, optional/alternative, and feature sets. Section 3: Creation of Models in a Model Chain provides modeling techniques for handling variation and applying variation in the functionality and behavior representations of the CPLSM.

2.3.1 Core Features of the CPLSM

The core features of a CPL are those that are mandatory for all components. Figure 2 illustrates the collection of capabilities and features that are required for all components in the CPL. Approaches for modeling features and variations that extend the core features are covered in Section 3: Creation of Models in a Model Chain, along with other specification modeling.

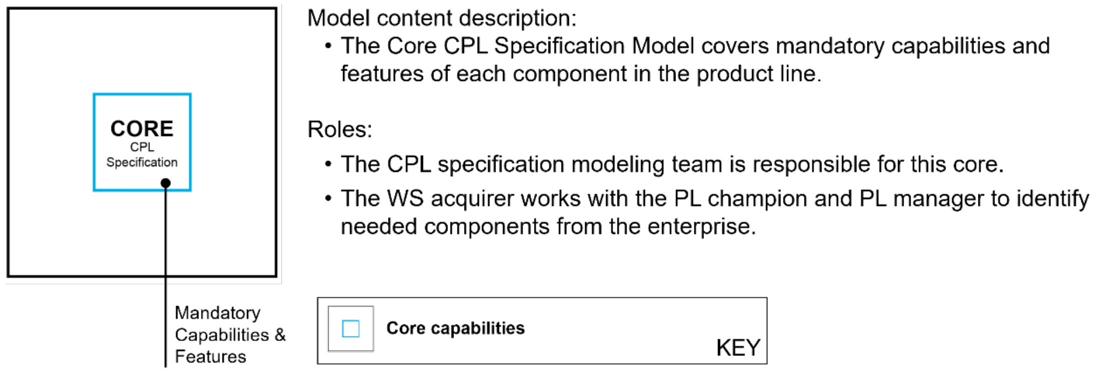


Figure 2: Core Features of All Components in the CPL

2.3.2 The 150% Feature Model

Variations to the core capabilities are identified as VPs via optional and alternative features within the features. The CPL specification modeling team is responsible for identifying and modeling these optional and alternative features. Figure 3 indicates that the optional and alternative features enhance the CPL core to support required capabilities across a planned set of use cases. Additional optional and alternative features may be added by a supplier, acquirer, or integrator for a specific component. Optional and alternative features apply at VPs within the core features. The total collection of features, including optional and alternative, constitutes the feature view of the 150% model. The term *150% model* reflects the fact that no single product will include all of the features in the feature model. A selected subset of features will constitute an instance model.

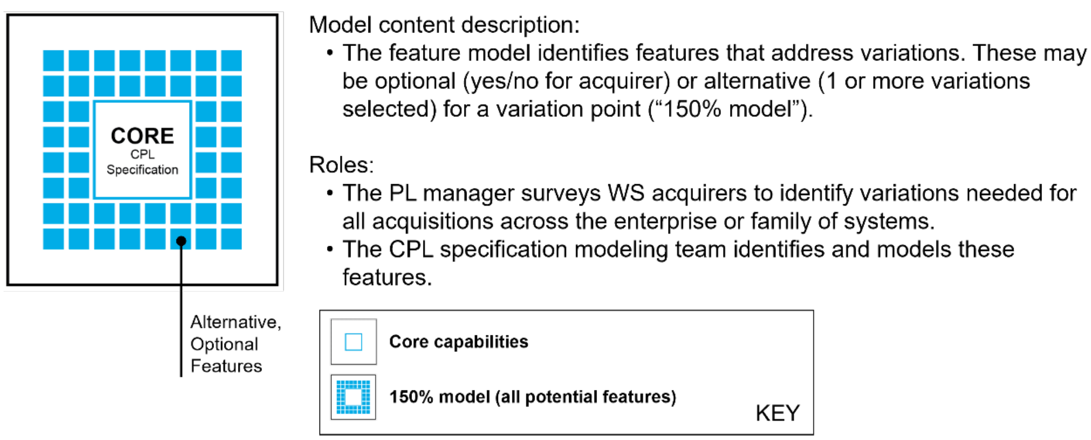


Figure 3: 150% Model—Features of the CPL Including All Options and Alternatives

2.3.3 Component Instance (100%) Model

The CPL specification modeling team will also model the features that comprise feature sets or combinations of features that will apply to a specific CPL component. The collection of features for a single component in the CPL is the 100% feature model or instance model for that component. Feature sets are derived from an understanding of the combination of required, optional, and alternative features that future WS users will most likely need.

The component instance modelers create a selection of feature variations—the core along with optional and alternative feature selections—to constitute multiple feature sets for the CPL. Figure 4 illustrates one selection of features for an instance model feature set within the CPL. A WS acquirer, supplier, or integrator may also create the instance model.

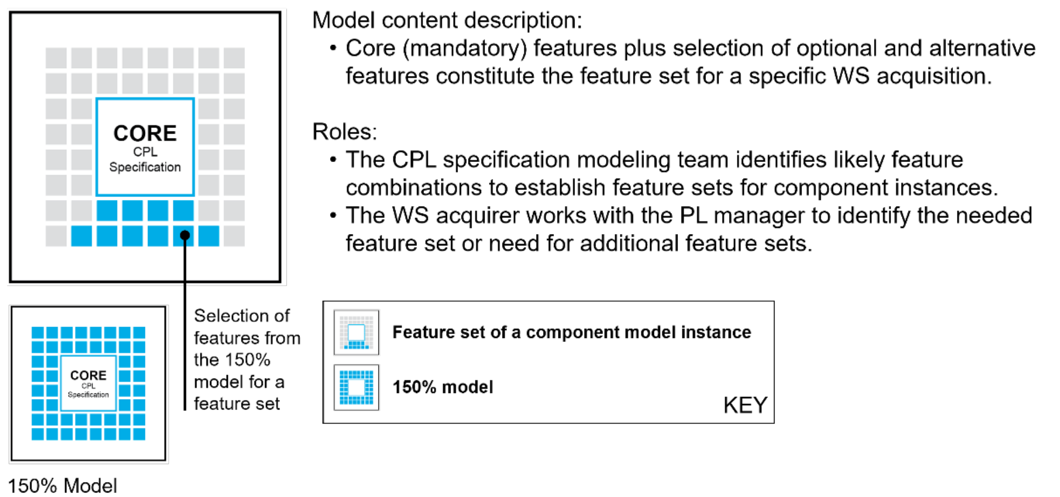
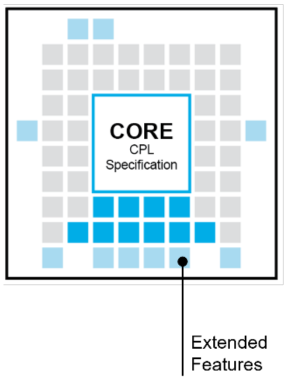


Figure 4: Component Instance (100%) Model—Selection of Core Features, Options, and Alternatives

Feature sets are used to determine what to deliver to the marketplace, as shown in Figure 1: Product Line Strategy: From CPLSM to Integration-Ready Components in the Marketplace. Suppliers who create a production capability for a CPL must consider any combination of features and variations. This combination covers the full range of the 150% model for potential feature set instantiation. This level of potential variation exists although only a subset of feature combinations will ever be instantiated for the marketplace for eventual use in WSs.

2.3.4 Model as Integrated

The WS will consist of components from the CPL along with those that are unique to the system. The acquirer and integrator use the models in the marketplace as a resource for evaluation of potential components for their system, select one or more for virtual integration and analysis, and then down-select to a component implementation for integration. The specific offered feature set of the CPL components may not satisfy the complete needs of the WS. In that case, WS acquirers and integrators must extend the feature set captured in the model or implementation, as shown in Figure 5: Weapon-Specific Instance Model Extended by Weapon System Integrator. New features not defined in the 150% model are shared with the CPL Organization to determine their suitability for sustaining the CPL. The Governance Report describes the decision framework for determining whether they should become part of the extended 150% feature model [Schenker 2023]. This governance scenario is covered in Sections 4.4: Scenario 2: Modify/Update an Existing Component Product Line and 4.5: Scenario 3: Application of Component Product Line to a Weapon System.



Model content description:

- Model instance specializes the CPLSM according to feature set variation selections.
- **Additional** features may be added to account for WS-specific needs.

Roles:

- The supplier produces the model instance for any or all feature sets. Instance models will be part of the component marketplace. The supplier may extend features covered beyond the feature set.
- The WS acquirer and integrator use the models in the marketplace as a resource for evaluation of potential components. The integrator may extend features covered beyond the feature set.

Feature set of a component model instance

Component model instance

KEY

Figure 5: Weapon-Specific Instance Model Extended by Weapon System Integrator

3 Creation of Models in a Model Chain

This section establishes modeling concepts for the CPLSM. It describes a model-based systems engineering (MBSE) process for developing components for the CPLSM. The process creates representations of the CPL under development that are required for description and analysis. As suppliers apply the CPLSM, they create representations to bridge the gap between representations of the CPLSM (functional, behavior, data, and features) and the concrete realization as executable implementations. This report does not address those CPL aspects addressed by suppliers.

The CPLSM representations capture specific aspects of the functions and behavior to be realized in instances of components within the CPL. The CPLSM also helps acquirers understand the suitability of the CPL for WS use. For component integration, the models provide guidance to integrators with respect to features that must be implemented, in particular for communications and connectivity. The CPLSM includes a feature model for capabilities needed by components of the CPL and functional models with VPs for those components. The model content helps developers focus on the use of features, feature sets, and VPs to specify variation across components of the CPL.

The representations chosen by the engineer may have predefined rules for including specific content in models. This report uses the term *submodel* to refer to representations included as part of an overarching model. This definition recognizes the modeling reality that no single artifact, picture, or specification is sufficient. Each is chosen to present a different perspective on the product being developed.

The modeling approach for CPLSM produces a series of submodels of the specification model. The collection of submodels constitutes a "model chain." Each submodel on that chain relies on information input from various sources, primarily other submodels of the model chain. Table 2: Stakeholder Roles lists information needs by stakeholder. These information needs are applied to specific submodels of a CPLSM as described in the modeling guidance in this section. The guidance includes a description of the submodels, their scope and modeling type, the qualities those models should address, and information provided to other submodels.

3.1 Scoping the Product Line

The CPL Organization working with the acquisition community develops a scope definition as part of the CPLSM that addresses needs shared across that community. They base this definition on needs that can potentially be met by an instance from a CPL. The scope will include

- a textual definition of the CPL
- use case models and activities
- feature models
- quality and deployment variations

3.1.1 Five-Part Textual CPL Definition

Table 3 gives the five-part product line definition from the Strategy Report [Cohen 2022, Table 3]. It can be used as a template to guide the definition activity.

Table 3: Informal CPL Definition—Used to Assess Need

Five-Part Product Line Definition: A software product line is …	Definition Applied to the Potential CPL
a set of software-intensive systems	describe the characteristics of the component needed by the WS
that share a common, managed set of features	list the capabilities and constraints that characterize the component
satisfying the specific needs of a particular market segment or mission	define the scope of use for components of the potential product line
and that are developed from a common set of core assets	based on CPL Organization experience, other components, tools, practices, etc. that can be applied to support satisfying the acquirer's need
in a prescribed way	provide the production plan for the potential product line and customization approach for it use

Describing the scope of use for components is a key factor in the decision process: Are there other WSs or potential uses of components if a CPL is developed? Is the number of potential WS users sufficient to warrant creation of the CPL? Are the variations in needs across the potential users understood? Can feature sets be identified to satisfy the needs of specific instance models of the WSs? These questions are the focus of the scenarios presented in Section 4: Scenarios, specifically Section 4.1: Scenario 1A: Create New Component Product Line—Determine Justification for Creating a New CPL and Section 4.2: Scenario 1B: Create New Component Product Line—Establish 150% Model for the CPL.

The WS specification represents the stakeholders' intent for the product, containing both functional and nonfunctional requirements. Stakeholders evaluate the requirements and develop test cases to assure that the right product will result from developing an implementation that adheres to the requirements. They also apply various techniques to evaluate satisfaction of nonfunctional requirements. The product line definition activity must consider these factors to assure that the CPLSM will provide the information that acquirers need to select or reject components from a CPL. If the definition is too broad or too narrow, the CPL may never support the marketplace. If components address too many areas of a WS, they cannot be tailored to address specific needs. If the definition is too narrow, the components will require other capabilities or have dependencies that the marketplace does not support.

3.1.2 Use Case Models and Activities

Component user or system interactions with the component can be represented as use cases using the typical use case model representation shown in Figure 6.

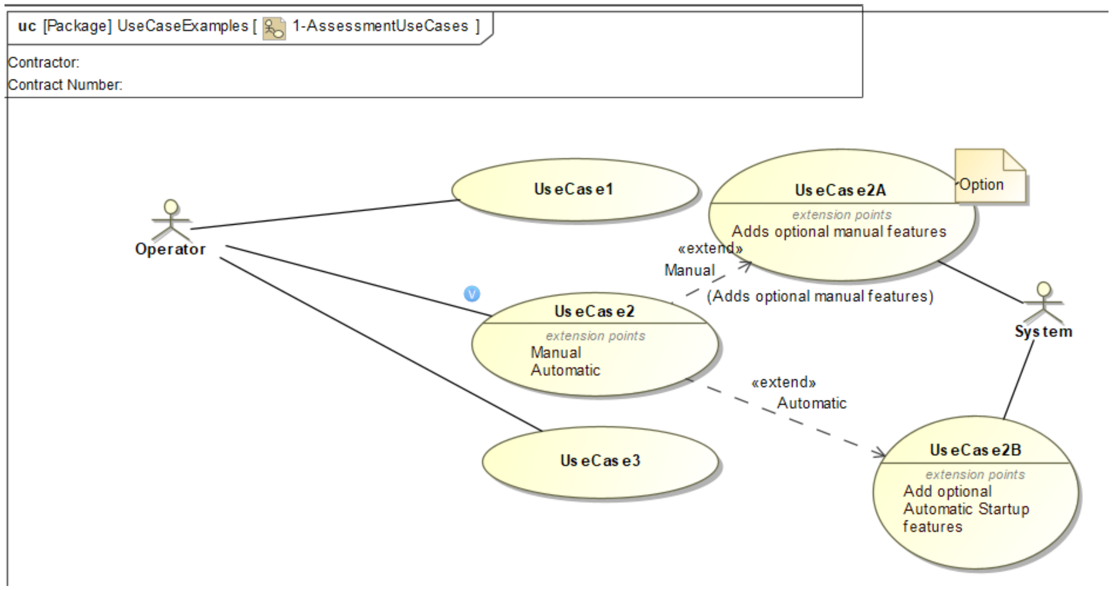


Figure 6: An Example Use Case Showing Optional Extensions

The example includes three use cases, and UseCase2 includes a VP for selecting one of two optional extensions. Each use case may include a description of the activities it encompasses. For the assessment, these may be a list of UseCase steps. For example, if UseCase2, a StartUp use case, shows the steps taken by a component on start-up, the variation may be for a user-controlled start-up or an automated start-up, as shown in Table 4.

Table 4: Steps Within UseCase2 Showing Variation

Step	User Controlled	Automated
1	Issue start-up	Perform built-in test
2	Read status	Read configuration file
3	Input start-up configuration	Connect via configuration

3.1.3 Feature Model or Table

The feature model for a CPLSM shows mandatory features that constitute the core for the CPLSM and those features that are options or consist of several alternatives. Product descriptions in the form of requirements, specification, data sheets for actual products, and industry standards can provide information about potential feature sets for a product line of components. A table or tree can represent the information captured from these sources as feature groups. Features within a group may be

- mandatory: Every product will include the feature. The collection of mandatory features constitutes the core features of the potential product line, as illustrated in Figure 2: Core Features of All Components in the CPL.
- optional: A product may include this feature in a feature set for a specific component, as illustrated in Figure 3: 150% Model—Features of the CPL Including All Options and Alternatives.

- alternative: At least one of the alternatives must be part of each feature set, but two or more alternative features may be selected. This category is also shown in Figure 3: 150% Model—Features of the CPL Including All Options and Alternatives.

Many representation methods exist for depicting the features in a feature model. The methods shown here are

- feature table
- feature-oriented domain analysis (FODA) tree
- MBPLE feature model

Each method is illustrated along with the representation it supports for the feature set.

3.1.3.1 Feature Table

A feature table shows internal and external features that provide a base-level component with product variants selecting among optional or alternative features [Felfernig 2022]. The example in Table 5 identifies the VPs that can later be captured in the architecture for externally visible features or in the design for VPs that are hidden behind an interface that can select among alternatives.

Table 5: Feature Model in Table Representation

Example	Feature Name	Base (Required)	Feature Set 1	Feature Set 2	Feature Set 3
Group A	A1 A2 A3	Yes No No	Yes Yes No	Yes No Yes	Yes Yes No
StartUp	User Automated	Yes No	Yes Yes	Yes No	Yes No
Group C	C1 C2 C3 C4	Yes No No Yes	Yes Yes No Yes	Yes No Yes Yes	Yes No Yes Yes

From Table 5, the CPLSM team can create feature sets that address the needs of specific, anticipated WSs that will use a CPL instance. The combination of features in Table 6 represents those that the integration-ready component will support for the WS.

Table 6: Two Feature Sets Selected from the Feature Model in Table 5

Feature Set 1: A1, A2, User, Automated, C1, C2, C4
Feature Set 2: A1, A3, User, C1, C3, C4
Feature Set 3: A1, A2, User, C1, C3, C4

3.1.3.2 FODA Feature Model Tree

Figure 7 shows information from Table 5 in a tree format. This representation uses the FeatureIDE tool set [FeatureIDE 2023], one of many based on the work of Kang, Cohen, and colleagues [Kang 1990]. It uses icons depicted in the legend as mandatory for core features as well as icons for optional and alternative features that encompass the 150% model. The legend entry for Abstract Feature indicates that this feature is organizational; Concrete Feature indicates those features that represent features of the CPL.

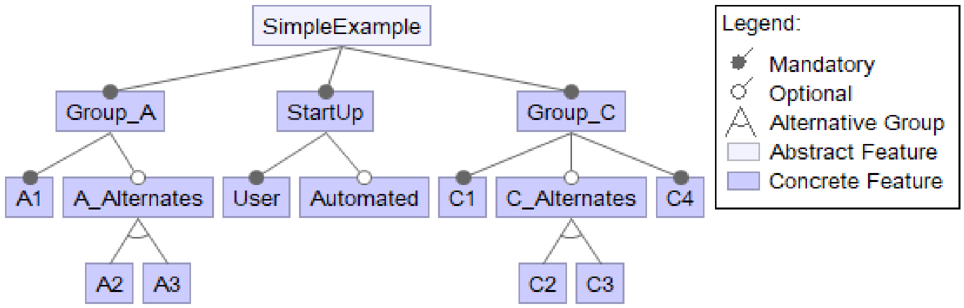


Figure 7: Feature Model Representation with Core and Variations

Figure 8 shows a selection of features from the Feature Model Representation of Figure 7. This feature set corresponds to Feature Set 3: A1, A2, B1, C1, C3, C4 from Table 6: Two Feature Sets Selected from the Feature Model in Table 5.

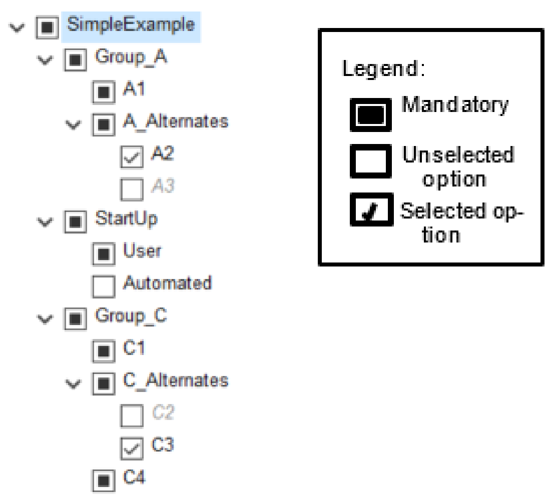


Figure 8: Feature Set of Desired Features Selected from Table 5

3.1.3.3 MBPLE Feature Model

Figure 9 shows a very simple example of an MBPLE feature model with the root feature and three feature groups corresponding to core features shown in the table view of Table 5: Feature Model in Table Representation and the FODA view of Figure 7: Feature Model Representation with Core and Variations. The StartUp feature has a mandatory mode feature that is a user-selectable option: User or Automated.

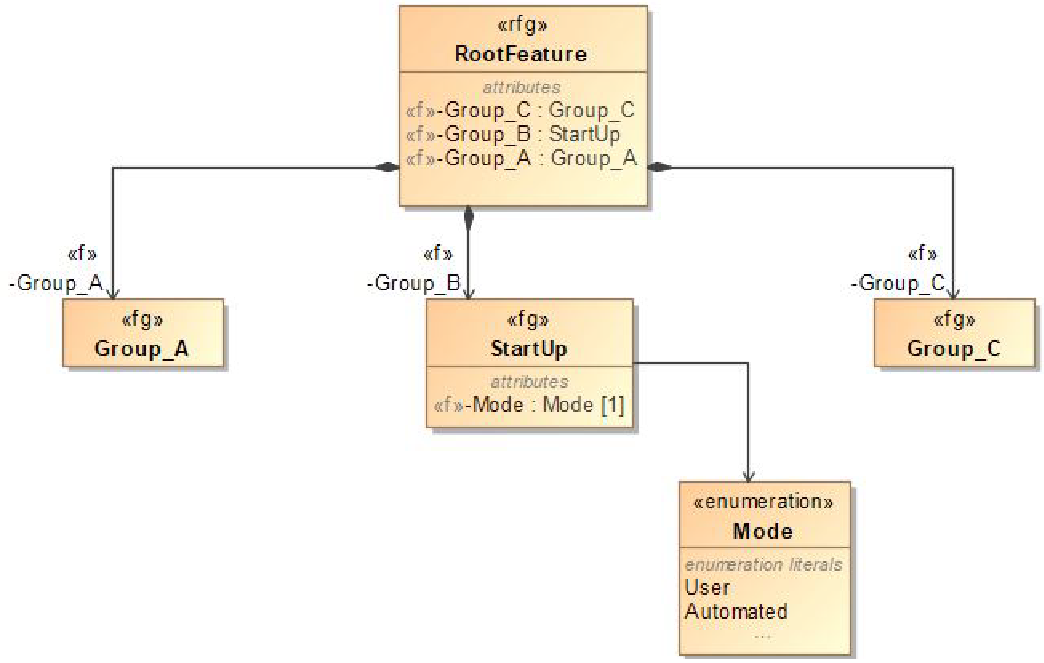


Figure 9: A Simple Feature Model in MBPLE

The acquirer or other stakeholder who wishes to create an instance of the RootFeature feature model must accept all three mandatory features and select one of the two options for mode. Figure 10 shows that the acquirer has selected the User option for the mandatory StartUp feature.

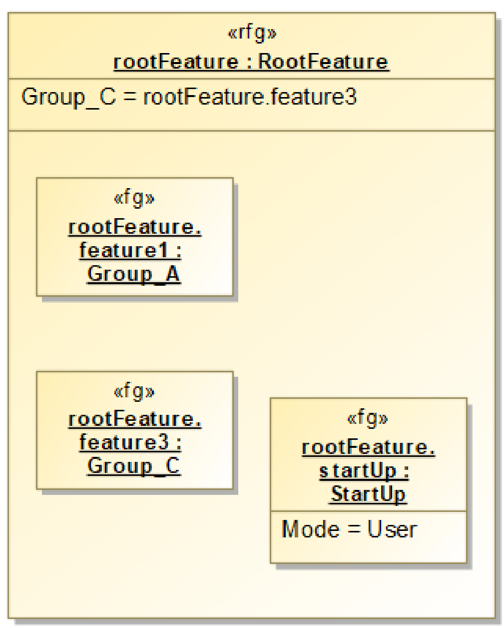


Figure 10: An Instance Showing Mandatory Features and Selectable Options

This selection of the core features with options and alternatives constitutes a feature set for the CPL consisting of Group A, StartUp, Group C, and the User mode option for StartUp.

3.1.4 Quality and Deployment Variations

The scoping of the CPL also examines features that constitute desired quality attributes or other nonfunctional requirements. These may include development needs such as architecture requirements to support extensions, modifications, or framework interfaces. The assessment should also include deployment characteristics that may determine whether the component is fully integrated into the WS or integrated across a network or other transport service. Start-up services may also be a characteristic of deployment, whether fully automated or under user control. Table 7 illustrates a selection of these quality attributes or nonfunctional features.

Table 7: Example Quality Attributes or Nonfunctional Features

Quality or Nonfunctional Requirement	Acceptable Ranges or Mechanisms
Memory size bounds	Required memory
Wait times	Assumed bandwidth and latency requirement
Processor requirements	Processor throughput requirements
Security	Mechanism: built-in security
Safety	Mechanism: built-in load status verification
Development needs	Driving architecture requirements (modifiability, extensibility)
Deployment strategies	Fully integrated, networked, start-up (user-controlled or automated)

3.2 CPL Specification Modeling for the 150% Model

The CPL scoping defines the boundary for the CPL. In the context of the modeling approach, the CPL specification modeling team builds on the scoping submodels to elaborate them into a 150% model. As defined in MBPLE, the 150% model employs MBSE practices to create a system model [MBPLE 2021]. For the purposes of the CPL, the 150% model addresses the CPL across its variations, defined by variation sets to create the CPL model. This report covers only the MBSE needed to model for the CPLSM. Suppliers, acquirers, and integrators use other techniques within MBSE to design and implement other CPL models as a 100% model. The 100% model and the integration-ready component are instantiated and integrated for a specific WS.

The full activity of creating the CPLSM might involve developing other artifacts such as component simulations—for example, creating instances that can be simulated in a modeling tool. These additional artifacts are for CPL evaluation purposes but are not covered in this report. The CPL Organization evaluates tool, method, and process alternatives to determine the best approach for each CPL. Table 3: Informal CPL Definition—Used to Assess Need includes each of these resources as core assets for the CPL and incorporates them into the production strategy identified in that table.

To establish the 150% model, the CPL Organization must build on the scoping submodels and proceed through a series of model development and evaluation activities. The CPL specification modeling team and appropriate stakeholders participate in that development. A CPL 150% model development effort creates model representations of the CPL for both development-related documentation and evaluation purposes:

- For development, the 150% model of the product line captures the full coverage of features. The CPL specification modeling team uses these features along with functional model approaches that incorporate VPs within those features—how elements of the specification will vary to capture not only the core features but also the variations driven by options and alternatives. VPs are also driven by quality and deployment features to satisfy the varying needs across the WSs that will be users of CPL component instances. The need for built-in variation is to accommodate the full range of WSs that will potentially integrate CPL components as they are developed in integration-ready form.
- For evaluation, the 150% CPLSM must allow potential users to determine if the available feature sets accommodate their specific needs. If the component cannot satisfy those needs, potential users working with the CPL manager must assess what further refinement is possible to create a new or extended feature set. Section 4.4, Scenario 2: Modify/Update an Existing Component Product Line establishes the need to minimize addition of VPs. Features and the variations they require should be identified as the CPLSM is developed, reviewed, and refined.

The full CPLSM includes the scoping submodels plus additional artifacts. This section deals with the need to establish modeling practices and evaluate the models through the development lifecycle. The content of the 150% model includes the artifacts above plus others. This section covers the concepts with examples for the submodels listed in Table 8.

Table 8: Submodels for the 150% CPLSM Artifact

• A context diagram showing an instance in the context of other system elements or human operators
• A list of technical requirements including functions, nonfunctional requirements, and architecture requirements: Architecture requirements specify how the CPL should be modeled in terms of standards, best engineering practice, and stakeholder directives for reuse of preexisting artifacts.
• Other requirements to cover development and delivery needs of acquirers, describing capabilities that users desire in the operational product
• Use case models and activities (expanded and refined from those of the scoping artifacts): These identify actors (users) and their expected interactions with the product. These interactions are elaborated in various forms such as scenarios or activity diagram flows.
• Feature models to establish required, optional, and alternative features of components in a product line: Features are product characteristics that include capabilities, user classes, nonfunctional requirements, and other areas to support definition of scope for the product line and definition of feature sets for individual instances.
• Quality and deployment variations
• Functional architecture (FA) to provide a hierarchy of functionality: The FA is based on decomposition and may include or imply data interactions. The FA is populated with VPs to show how functionality differs based on feature selection.
• Sequence diagrams and state diagrams with VPs to elaborate use cases and functionality
• Standards documents: These may be industry or government standards that require product conformance evaluation and set specific expectations of the product for a user.
• Data architecture: This architecture supplies the data definitions and higher level abstractions that characterize interactions and connections between components.

The model chain includes this collection of submodels for scoping the full CPLSM. Fundamental questions that should be asked early and often during refinement and creation of the 150% model include the following:

- Why is this model being created?
- Who will use the model?

- What other models or submodels will be derived from this model's contents?
- When should each model be created?

3.2.1 Model Chain Evolution

Figure 11: Model Chain Providing Expanding CPL Content illustrates the sequence of model creation and refinement necessary to specify the 150% CPLSM model and to apply that model in the context of a model instance. The succession of product models is represented by CPLSM Versions V1, V2, and V3. Across each version, the specification modeling team adds content to models in a model chain.

- Component Model V1: The contents of the CPLSM reflect the specification content, with supporting submodels to cover functional (L1) and architecture (L2) requirements content. Another submodel includes the feature and feature variations needed by components across the CPL.
- Component Model V2: This version of the CPLSM refines the requirements and feature information to an FA with VPs, provides full requirements specification, and defines behavior—the activity or state diagrams associated with use cases. Component Model V2 is the 150% model delivered by the CPLSM team to suppliers, acquirers, and integrators. For this example, additions of submodels (L3 and L4) add more specification content. The content may be functions, performance, behavior, or another type, depending on the role using that version of the product model.
- Component Model V3: This model represents more submodel detail (L5) added by a supplier or an integrator. Model guidance for this content is not included in this report; however, the CPLSM submodels provide input to V3. Understanding how other submodels in the model chain use content from the CPLSM is essential for guiding the specification modeling team.

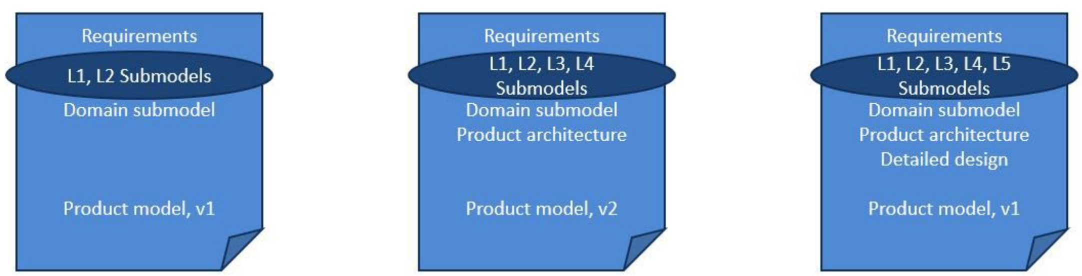


Figure 11: Model Chain Providing Expanding CPL Content

3.2.2 CPLSM Modeling Plan

The modeling strategy for a CPL includes a modeling plan that specifies a sequence of submodels to be built by the stakeholders in the product line. The submodels fall into categories such as specification, design, and implementation. The artifacts are created using multiple representations of information needed to accurately construct other submodels as well as implementations. The plan described in this report covers only those for CPL specification. The scoping submodels are the first link in the model chain including domain models, use cases, requirements, and others. The later models created for the 150% model have traceability and derivation relationships with earlier models. They provide a “chain of evidence” to boost confidence in predictions of fully identifying features, feature points, requirements, FA, and behavior to guide design and implementation.

The modeling plan reflects content that should be captured in the CPLSM. The plan establishes the timing of adding each new link (or submodel) to the chain. It also describes the attributes of each model needed to assure that planned analyses (e.g., correctness, completeness, consistency) can be carried out. The specifics of the modeling approach and tool support are major determinants of which submodels will be needed.

The number of models in the chain and the exact content of each type of model are defined by the development process. Additional diagrams and other artifacts are created extemporaneously to assist during informal design discussions. These models are obviously not part of the a priori modeling plan. If the informal model proves to be sufficiently useful, these models can become part of the complete CPL model chain as it evolves over time. Developing planned or ad hoc submodels consumes considerable engineering resources since the artifacts must be created and, in many cases, sustained.

The type and number of submodels that will be necessary and sufficient depends on the specific modeling situation and the purpose of modeling for that situation. Many submodels will be created just to validate or reject a proposed design. The modeling plan should give clear guidance on criteria to use in determining which submodels to retain and which to discard. Models of rejected design decisions may be of use as documentation and lessons learned just as much as those submodels defined in the modeling plan.

Models exist to support decision making, tradeoff analyses, and documenting the specification of the CPL. A model must be constructed using a representation with sufficient semantics to express the attributes needed to support these activities and to reason about the relationships among components. To be meaningful in the broader development context, the modeling plan provides criteria to determine if a given submodel is

- **complete within the current context.** Not every potential instance that can be created from the 150% model will be modeled and analyzed, but the model should be evaluated to determine that it can support random selection of features to create an instance within a specified context.
- **unambiguous.** Each element in the modeling language must have clear semantics. Using appropriate tools for the artifacts listed in Table 8: Submodels for the 150% CPLSM Artifact provides that assurance.

Requirements for a WS capability may evolve and diverge from the original set. An MBSE approach that provides mechanisms for analyzing early and ongoing representations of the 150% model can determine whether that model is sufficiently flexible to accommodate the changes. The WS acquisition program should continue to collaborate with the CPL Organization to inform the CPLSM team developing the 150% model. By organizing the 150% model using the preset submodels, sometimes called the metamodel, the CPL model chain can be tailored to the individual using the model. Figure 12 depicts a metamodel for a CPL modeling plan. This report concentrates on the leftmost branch of the metamodel: the Component Specification Model.

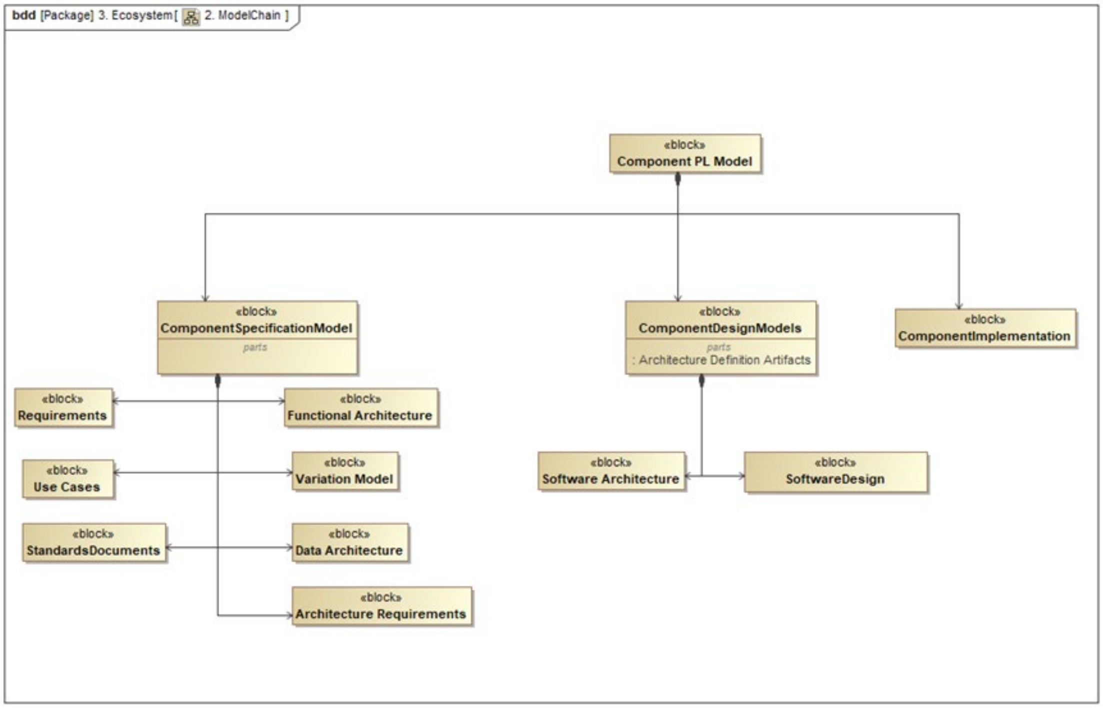


Figure 12: Categories for Component Models and Subordinate Submodels

3.2.3 The 150% Model Artifacts

This section provides concepts and modeling guidance for the 150% model. It elaborates on the scoping submodels and introduces submodels not already covered.

3.2.3.1 Context Diagram

The context diagram places the CPL in the context of other components, systems, or users. That context is a source of variation, so the context diagram may be annotated with VPs. Its main purpose is to provide the "setting" for the component. Figure 13 represents a context diagram for the StartUp Component example. It shows the VP for automated or user features. It also shows features to Find Identification of Network Devices (FIND) and obtain status on start-up.

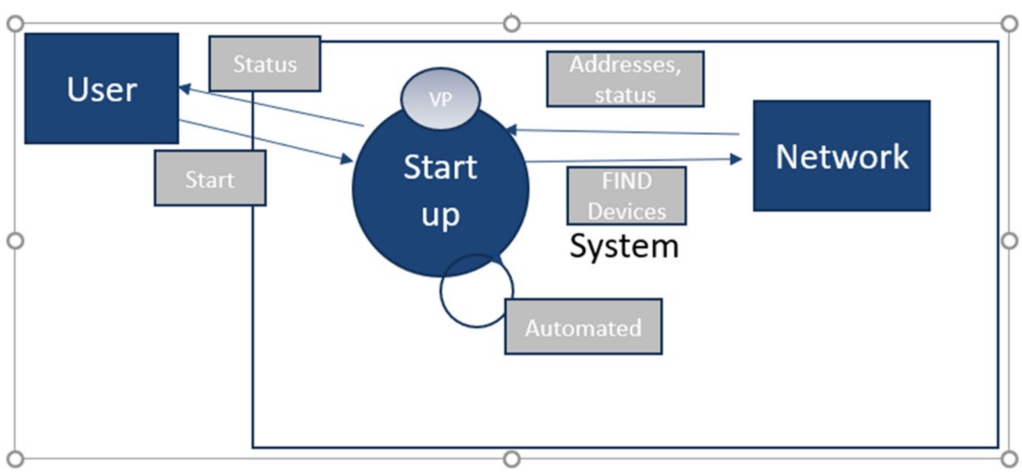


Figure 13: Context Diagram Example for a Start-up Function or Service

3.2.3.2 Requirements

Requirements may be in table or diagram form and encompass both functional and quality attribute needs and descriptions. A requirements table may include the name, description, and rationale for a requirement. It should also cite the source for the requirement, especially where an industry or government standard is applied. The table is usually represented in an outline fashion, broken down to requirement groups, subgroups, and individual requirements with test cases. Modeling tools also support diagram notation for requirements. These generally follow a tree structure suggested by the tabular outline.

3.2.3.3 Use Case Models and Activities

A use case model describes (visually) the proposed functionality of a system or product. A use case represents a discrete unit of interaction between a user (human or machine) and the system. This interaction is a single unit of meaningful work, such as InitializeDevice.

Each use case describes the functionality to be built in the proposed system, which can include another use case's functionality or extend another use case with its own behavior.

A use case description will usually include

- general comments and notes describing the use case
- a reference to requirements: the formal functional requirements of things that a use case must provide to the end user, such as <ability to start device>. These correspond to the functional specifications found in the requirements artifact and form a contract that the use case performs some action or provides some value to the system.
- constraints: the formal rules and limitations that a use case operates under, defining what can and cannot be done. These include
  - pre-conditions that must have already occurred or be in place before the use case is run; for example, <create order> must precede <modify order>
  - post-conditions that must be true once the use case is complete; for example, <order is modified and consistent>
  - invariants that must always be true throughout the time the use case operates; for example, an order must always have a customer number
- scenarios: formal, sequential descriptions of the steps taken to carry out the use case, or the flow of events that occur during a use case instance. These can include multiple scenarios, to cater to exceptional circumstances and alternative processing paths. These are usually created in text and correspond to a textual representation of the sequence diagram.
- scenario diagrams: sequence or activity diagrams to depict the workflow and similar to scenarios but graphically portrayed
- additional attributes, such as implementation phase, version number, complexity rating, stereotype, and status

3.2.3.4 Standards

Standards may include documentation, industry, government, and organizational standards. Project guides must also be cited as authorities. The level of authority may span the enterprise, product family, or development organization. An organization may develop a metamodel to specify the models and submodels that the modeling team should create. Figure 12: Categories for Component Models and Subordinate Submodels offers an exemplar to follow for developing the 150% model and for the modeling approach to create individual submodels or reuse specific models.

3.2.3.5 Feature Model

For the 150% model, the features reflect a complete collection of functions, constraints, and deployments, and creating features in the 150% model follows the same approach as defined for the assessment artifacts in Section 3.1.3.3: MBPLE Feature Model. The approach for the 150% model introduces a complete MBPLE approach for representing the features. This approach applies the same concepts as during CPL scoping (Section 3.1.3: Feature Model or Table), but with different rules for identifying each feature group. The feature model begins with a RootFeature. Under that RootFeature are FeatureGroups that may be at one or more hierarchy levels. Under the FeatureGroup are individual features that must be enumerated to allow for individual selection to create the instance or 100% model needed by the acquirer or other CPL user.

Mandatory features appear in every product. The collection of those features constitutes the core features of the CPL. While the features are mandatory, a mandatory feature may offer to the user sub-features that are options (one choice out of many enumerated features) or alternatives (one or more choices of enumerated features). Rather than indicating the extension of a mandatory feature though a specific symbol, the MBPLE feature model uses enumeration literals: 1 for the choice of an option where one of two or more must be chosen and 1..*n* where 1 to *n* features can be chosen. For example, an optional feature for a start-up sequence may be user-operated or built-in when powered on. The error response to a start-up could be a flashing light, sound cue, automatic restart, or others. The instantiation requires a feature model user to select one or more of these alternatives.

3.2.3.6 Functional Architecture (FA)

An FA can be captured with the Systems Viewpoint (SV) SV-4 Systems Functionality Description [DoDAF 2010]. The SV-4 can be depicted as a tree structure that decomposes the component into functions and shows decomposition of functions themselves, if necessary. An alternative form also shows functionality in terms of activity and data flows between resources. However, the SV-4 does not show runtime behavior, control, and interactions among the different entities. Figure 14 shows an extract of the Data Loader FA SV-4 as an example of this diagram for a hierarchical functional decomposition of a product application. The level of detail is consistent with the level of detail in the use case description. However, depending on the analyses to be done for correctness, completeness, and consistency, the view may need more detail. Also, developing relationship tables between functions and features or between functions and requirements may drive the need for greater detail.

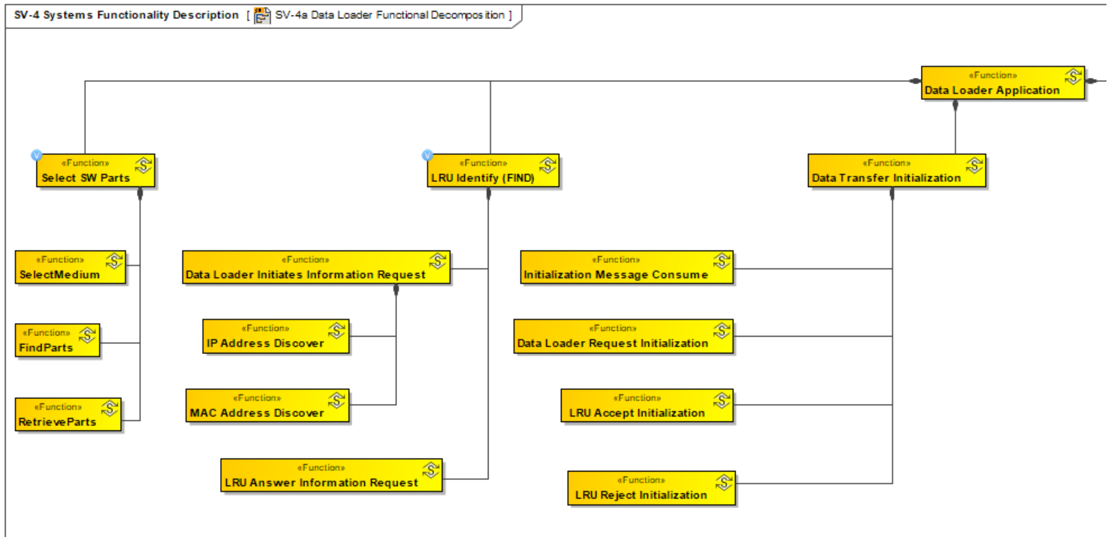


Figure 14: Use of SV-4 to Represent Functional Architecture (Hierarchy)

The circle-Vs in the figure on the functions "Select SW Parts" and "LRU Identify" indicate VPs. Software part variation covers the medium on which a part is stored, such as CD or USB. The selection of the specific feature in a feature set indicates the processing to retrieve that part. The variation on LRU Identify can be a deployment alternative. The data loader may connect to a system network directly or through an interface that is independent of network type. That alternative deployment feature drives different designs for the LRU Identify function.

SysML block diagram notation can also support the description of FA. This notation can include data flows between blocks.

3.2.3.7 Behavior

An activity flow description shows flow interactions between functions identified in the FA. To achieve representation of behavior flow, the Services Viewpoint (SvcV) SvcV-4 Services Functionality Flow Description is an optional submodel diagram that may be included among the specification submodels (illustrated in Figure 15: SvcV-4 with Flow Description to Represent Logical View).

SysML state diagrams (Figure 32: SysML State Transition Equivalent of Figure 31) capture the behavior of a component that goes through different modes (start-up, operations, error recover, etc.). SysML sequence diagrams (Figure 34: Timeline in Sequence Diagram Showing the Upload Loop) are also used to represent component behavior in CPLSM. SysML activity diagrams are another representation that can capture CPL behavior.

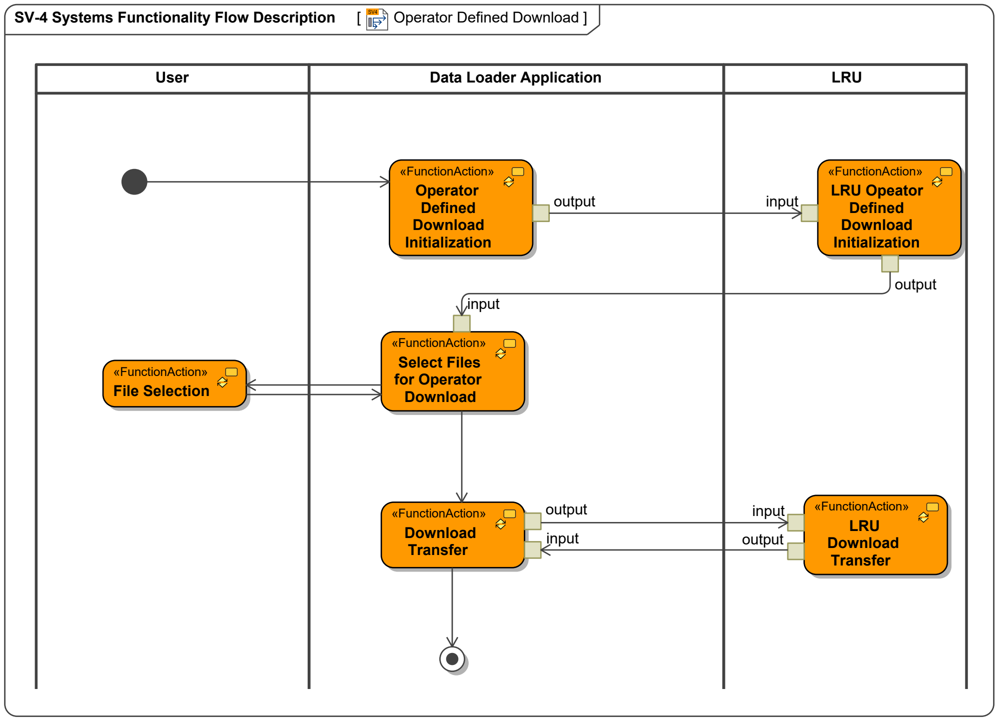


Figure 15: SvcV-4 with Flow Description to Represent Logical View

3.2.3.8 Data Architecture

A data architecture includes guiding specifications, governance policies, and key WS area data abstractions with the primary purpose of supporting an interoperable means of data exchange among software components. The architecture supports the rules that guide the conceptual basis for data definition and data logic captured by the abstractions. The data model applies the architecture to select the data concepts and logical abstractions that define data structures and elements within those structures, are shared by components as state abstractions, or define flows between components. Data models help represent specific data constructs that are required and the format to be used for common application across a single or multiple CPLs. A physical data model uses specific data types for component implementation, integration, and exchanges across the weapon system.

The modeling may include

- structures, including abstractions and the Domain-Specific Data Model (DSDM)
- data representation
- data storage and persistence
- data in transit

Many WSs are dependent on a common operating picture (COP). Concepts define data that represent elements that the COP will track and the logic for abstractions that assemble those elements for management and state sharing across components of the WS. In addition to describing the data needs, the intent of the data architecture is to define the data needs, the semantics, and the logical abstractions in an unambiguous fashion. Block diagrams in SysML are frequently used to represent the data models that constitute the data architecture. Further refinement of those models provides the conceptual and logical representations of those abstractions.

According to the *Data Management Body of Knowledge* [DMBOK2 2017], data architecture defines the blueprint for managing data assets by aligning with organizational strategy to establish strategic data requirements and the data model designs to meet those requirements. DMBOK2 defines data modeling as "the process of discovering, analyzing, representing, and communicating data requirements in a precise form called the data model."

3.2.3.9 Architecture Requirements

An architecture requirement describes a key system property (e.g., quality attribute) that has an important role in determining the architecture of a system and may include an appropriate pattern, process, standard, tool, or guidance to meet the desired attribute [Padilla 2019]. The CPL is guided by these requirements, which include the specific models needed for the CPL definition, adherence to standards, data architecture, and verification methods. As with functional requirements, the architecture requirements may be represented in tabular or diagram form. The table will include the requirement name, text description, source text, and rationale. The table may include links to related architecture requirements. These requirements are also expressed in an outline, numbered fashion. A supporting diagram depicts the requirements in a tree structure with relationships mapped into the diagram. Examples of an architecture requirement name and description include the following:

- Maintain a CPL architecture: Establish and manage a system architecture for the CPL to implement the Modular Open Systems Approach (MOSA) Family of System objectives.
- The CPL shall leverage one or more DSDMs: System interfaces for the family of systems modeled against a consistent DSDM provide a mechanism that can streamline and potentially automate integration and test actions. The organization's DSDM will eliminate semantic ambiguities presented when integrating with separate and disconnected data models.
- A system's component interfaces shall enable implementation variation: Points of extension of function should be traceable to the core (FA) specification and the feature model. The documentation (in model form as a possible mechanism) should identify and describe which interfaces offer increases in functional capability.

The architecture and technical requirements guide model content, the modeling plan, and relationships between models or submodels. An example is the relationship between the FA and the feature model. They are also applied in architecture or design evaluations to assure that each architecture requirement is consistently and correctly applied.

3.2.4 Model Review and Analysis

Models undergo a variety of reviews, evaluations, and analyses during the development and across the model chain. The purpose of these activities is to objectively determine the following:

- Each model achieves the quality factors of completeness, correctness, and consistency. Organizational standards or project guides (see Section 3.2.3.4: Standards) will include criteria for these evaluations.

- Models in the model chain are reviewed and analyzed to verify that model information and analysis results can be used by other models in the chain that depend on that information. For example, if a specification establishes a required attribute range, an architecture model can rely on that range of attribute values, or a subset of that range, as bounds for its content and analysis.
- Individual submodels of a single model exhibit consistent information to support sharing of data, functionality, behavior, and state transitions as appropriate for the overarching model. Development reviews and walk-throughs will assure this internal consistency.

As models increase in complexity and requirements increase in number, automated or semiautomated tool support is essential. Many tools check for syntax and semantic consistency—data provided by one model and used by another can be traced and verified for consistency. A function may be defined and used or included in a separate model with built-in checks on syntax and semantics. A broader set of requirements places an additional burden on model review and verification. These are the architecture requirements that define not only what capabilities or data the model must provide and use but also the quality attributes of how that capability or data is provided. Such architecture requirements may be similar to the following:

- All data definitions shall be based on a specified data model, either by direct use or by subtyping mechanisms.
- Capability A shall be modeled using an instance of the XYZ CPL from the component marketplace.
- Any communication between separately modeled components must interface through a single data transport mechanism.

Model analysis tools should be provided to engineers to perform the validation of these requirements or, at a minimum, to assist in determining satisfaction.

4 Scenarios

This Modeling Report incorporates scenarios developed in the CPL Governance Report to highlight modeling approaches for all sides of CPL component specification modeling, instance modeling, and component acquisition. The governance scenarios are tailored to emphasize specific aspects of the modeling approach across these modeling and acquisition needs. To illustrate the modeling aspects of these scenarios, this section presents a comprehensive exemplar to illustrate the content of each submodel in the model chain in terms of scoping, 150% model specification, architecture requirements, and guidance for instance tailoring.

This section of the Modeling Report presents the Governance Report scenarios from the perspective of CPL specification modeling. From the Governance Report,

> *These scenarios are specifically chosen to illustrate aspects of the product line lifecycle where governance may be an issue. Collectively, the scenarios should logically lead to a coherent set of activities, or processes, that will need to be established to enable a successful product line approach.* [Schenker 2023]

Each scenario listed in Table 9 is exercised as part of the Modeling Report to establish practices related to CPL specification modeling and related activities.

Table 9: Governance Scenarios Applied in Modeling

Scenario #	Description	Modeling Practices	Use of Resulting Models
Scenario 1A: Create New Component Product Line—Determine Justification for Creating a New CPL	Create New Component Product Line A. Determine justification for creating a new CPL.	Model scope to capture information for downstream specification models: features, potential feature sets, use cases, scope definition.	Guidance for creating the 150% model, including VPs within functional modeling and for deployment variation Identification of potential preexisting components and variation sets
Scenario 1B: Create New Component Product Line—Establish 150% Model for the CPL	Create New Component Product Line B. Establish 150% model for the CPL.	Develop primary CPL specification including feature, FA, and data architecture models. Optionally, the CPLSM may include behavior and state submodels. Each model includes identified VPs refined to feature sets with the ability to extend a model.	CPLSMs to be used in Scenario 2, to assess state of a component with respect to future use of that component as part of the CPL, and Scenario 3, for specializing and extending CPL models for a specific WS use Basis for CPL design
Scenario 2: Modify/ Update an Existing Component Product Line	Modify/update a legacy CPL.	Create CPLSM to effectively manage need for and scope of the change. Optimize model elements to identify VPs to minimize need for model change via selection of variations.	Understanding where the product line component is used within the enterprise, what VPs and features are being used by each platform, and other aspects of the change relevant to the platforms
Scenario 3: Application of Component Product Line to a Weapon System	Apply a CPL to a WS.	Refine 150% model to 100% instance model for a WS component.	Evaluation of 150% model plus variation sets to satisfy needs of the WS

The text used in these Modeling Report scenarios is tailored to better establish specific modeling practices. Specifically, Governance Scenario 1 explores the creation of a new CPL. The Modeling Report splits that scenario into two parts: The first part (Section 4.1: Scenario 1A: Create New Component Product Line—Determine Justification for Creating a New CPL) forms the information to justify the need for the CPL using the scoping submodels. The second part (Section 4.2: Scenario 1B: Create New Component Product Line—Establish 150% Model for the CPL) illustrates development of the 150% CPLSMs.

4.1 Scenario 1A: Create New Component Product Line—Determine Justification for Creating a New CPL

Scenario 1A in this report applies when a WS acquirer or multiple WSs require or wish to take advantage of the product line approach. The product line organization must determine feasibility of a CPL to satisfy needs across the scope of those WSs. The scenario consists of three steps:

1. Assess the need for a CPL.
2. Assess the marketplace for availability of preexisting components to include in the CPL.
3. Create component specification information for definition in a CPL.

4.1.1 Assess Need for CPL

4.1.1.1 Scenario Preconditions and Activities

The scenario has the following preconditions:

- The acquisition program for a WS identifies a new required capability that may be provided as a CPL.
- The WS program works with the CPL Organization to establish a list of those components that are already product line components.
- The CPL Organization coordinates with the WS program to fulfill its needs with the product line organization. Coordination with the WS is necessary to keep the acquirer informed of whether a CPL or specific component from the CPL will be available within the WS schedule.

The scenario activities are carried out by the acquirer and the CPL Organization, producing a set of artifacts:

A. The acquisition program queries the CPL Organization to determine if the WS need will be satisfied by an available product line component specification.

B. The WS acquirer collaborates with the CPL Organization. Together, they perform specific tasks in support of the WS acquirer and the overall enterprise:

   1. Assess CPL specifications that may satisfy the new WS capability via a product line component. The assessment looks at

      a. commonality across enterprise platforms

      b. existence of similar commercial off-the-shelf (COTS) products in the marketplace

      c. specifications of related, existing components

2. Develop a draft set of technical requirements for a CPL. Use the draft requirements to determine if legacy components or a CPL already exists. They can also serve as a basis for querying interest from other acquirers across the enterprise and alert the community of need.
3. Use the draft as part of a request for information (RFI) to test the marketplace for legacy component offerings.
4. Decide whether to incorporate this potential component into the set of product line component specifications or to establish a new CPL specification.

4.1.1.2 Artifacts of the CPL Evaluation

The assessment artifacts of the scenario activities form the basis for the go/no-go decision on creating the CPL. These artifacts include

- paper or briefing that provides the outcome of the evaluation (model and text of scope OV-1 and AV-1)
- rationale for the decision on how to proceed
- informal CPL specification provided to the acquisition program. This specification provides features and variations of a potential CPL and supports the decision-making process about whether a feature set and component from the proposed CPL will meet the acquirer's need.

As a starting point in assessing the need, the CPL Organization and acquirer develop a draft definition of the product line. They base this definition on the component that the acquirer needs and that can potentially be supplied by an instance from a product line. The CPL Organization applies the scoping submodels, as defined in Section 3.1: Scoping the Product Line, to support a go/no-go decision on pursuing a CPL strategy to address the WS needs. These submodels include

- the five-part definition of the product line, which is used as a template to guide the assessment (Table 3: Informal CPL Definition—Used to Assess Need)
- use cases (Figure 6: An Example Use Case Showing Optional Extensions)
- feature model (Figure 7: Feature Model Representation with Core and Variations)
- quality attributes (Table 7: Feature Model Representation with Core and Variations)

4.1.2 Assess Marketplace for Existing Components to Satisfy the CPLSM

The CPL Organization assesses or surveys the marketplace (the built-component supplier side of Figure 1: Product Line Strategy: From CPLSM to Integration-Ready Components in the Marketplace) to identify components that would potentially serve as a starting point for the product line. Here is an example:

- A preexisting component might incorporate a superior approach to providing the component's functional capability.
- The preexisting component might not achieve, or might minimally achieve, nonfunctional attributes identified in the survey of need, such as
  - memory size bounds
  - wait times in terms of latency, bandwidth, or schedulability
  - available processing power based on typical processor throughput limits
  - airworthiness for security and safety

- developmental needs for modifiability, extensibility, and other quality attributes

- Result: Consider using this component as a starting point for a component in the CPL.

Adapting components from legacy carries certain limitations:

- A supplier that has produced an existing component may be averse to modifying it as needed to support the product line approach. Modification may require the CPL Organization to award a separate contract to an organization willing to adapt and support the component as required.
- Intellectual property issues may emerge whenever multiple companies are involved in collaboration, and these should be worked out as part of the contracting process.
- Refinement and evaluation of the CPL might involve developing component simulations to be used for evaluation purposes. There might be a CPL evaluation facility that supports the assessment. The CPL Organization may also perform minimal modeling to support analysis of the above nonfunctional attributes—memory size, wait times, processing power, safety, security, and others.
- The CPL Organization assesses available alternatives for using preexisting components, identifies additional preexisting component candidates, considers the possibility of a product line production capability, and determines the best approach for each component and the CPL potential.

Scenario 1B expands on each of these artifacts if a CPL approach is viable. A CPL specification modeling team under the CPL manager will create more formal versions in addition to functional, behavior, and state models (where applicable) to create a 150% model for further application.

4.1.3 Scenario 1A Example—The Data Loader CPL

This report uses a data loader as an example component to illustrate Scenario 1 and to highlight aspects of the CPL specification modeling approach. A WS acquirer or multiple WSs may require a data loader and wish to take advantage of a potential Data Loader CPL. The data loader supports operation of a typical network-based system where other components—line-replaceable units (LRUs)—are connected to an embedded processor via a network. Each LRU runs specific software that must be loaded, and frequently reloaded, to support normal system operations. These systems use a data loader to bring in loadable software parts from some storage source, process those parts, and dispatch them to the correct LRU. The acquirer for the Scenario 1A example, and most other potential users, expects the Data Loader CPL to conform to the ARINC-615A standard that defines data loaders [ARINC 2007]. This expectation becomes a product line requirement for the component.

The preconditions of the scenario include identification of the need for a data loader, collaboration of the acquirer with the CPL Organization to clarify specific needs such as application of industry standards (e.g., ARINC), and coordination of CPL Organization status with the WS need. The scenario establishes a set of artifacts as products of the assessment:

- a context diagram for the CPL showing an instance in the context of other system elements or human operators (Figure 16: The Data Loader Context Diagram)

- a list of technical requirements including functions, nonfunctional requirements, and architecture requirements. Other requirements may cover development and delivery needs of acquirers (Section 4.1.3.2: Technical Requirements).
- CPL five-part definition (Table 10: Data Loader CPL Definition)
- use case models and activities (Figure 17: Use Cases from the Data Loader Assessment)
- feature models (Table 11: Data Loader Product Features, Figure 18: Feature Model Showing Groups of Features, and Figure 19: Expansion of the Software Feature Group)
- quality and deployment variations (Table 12: Marketplace Survey of Nonfunctional Requirements for the Data Loader CPL)

4.1.3.1 Data Loader CPL Assessment

The CPL Organization may already support the WS data loader requirement and will collaborate with the WS to determine if any instances are integration ready for the WS. If not, the CPL Organization will perform the scenario steps identified in Section 4.1.1.1: Scenario Preconditions and Activities. Figure 16 illustrates an example of the artifact to capture results of the assessment. It provides the context for the data loader operations where Radar and Transponder are examples of the LRUs.

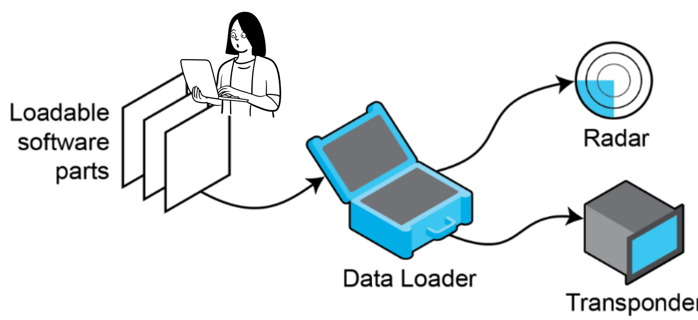


Figure 16: The Data Loader Context Diagram

The tasks illustrated by the context diagram are the interactions between the data loader, an operator, software parts for loading, and network devices such as a radar or transponder. Interactions with the environment support reading software parts and delivering software parts to designated target hardware. The data loader parses contents of those parts to determine the destination and content to be loaded. It communicates with the environment according to a network protocol. This graphic may form a portion of the DoDAF OV-1 model of a data loader component in a broader systems context [DoDAF 2010].

4.1.3.2 Technical Requirements

The scenario steps also call for developing a draft set of technical requirements. For the example, the ARINC-615A standard provides those requirements [ARINC 2007]. In addition, the CPL must account for architecture requirements that drive future design decisions. These may include the need for qualities such as extensibility and modifiability. They may also include conformance to the Future Airborne Capability Environment™ (FACE™) Technical Standard 3.1 [FACE 2020] such as the following:

- The CPL shall use the FACE Platform Data Model to represent software interface data.
- The CPL shall use the FACE Data Architecture to represent software component data.

Requirements may also include directives for deployment in a broader CPL context where other CPLs provide data loader capabilities:

- The CPL shall incorporate COTS, government off-the-shelf (GOTS), and non-developmental item (NDI) system components in the system architecture when an appropriate COTS, GOTS, or NDI system component exists.

CPL requirements may also include development directives that are necessary for WS acquirers to evaluate component suitability:

- The performer shall deliver final complete model specification.
- The performer shall deliver final model analyses.
- The performer shall specify all key interfaces.
- Performer deliverables shall enable government or third-party replication of the analysis results.

A marketplace assessment will examine vendor solutions that provide data loader capabilities. While the solutions address data loader functionality, they may not support an enterprise's guiding architecture requirements. For example, do vendor solutions integrate logging, health management, network management, and other capabilities that the WS acquirer desires as provided by services from individual product lines? This system architecture requirement may be driven by the need to satisfy the MOSA that is described in the Strategy Report. It may also be considered a deployment feature: How will variations in context affect deployment of a data loader component instance within a WS?

Finally, the CPL Organization will develop a draft characterization of the Data Loader CPL in terms of the five-part product line definition shown in Table 10.

Table 10: Data Loader CPL Definition

Five-Part Product Line Definition: A software product line is …	**Definition Applied to the Data Loader CPL**
a set of software-intensive systems	software and hardware (optional) to process loadable software parts for delivery to networked LRUs
that share a common, managed set of features	storage media for parts, upload and optional download modes, obtainment of LRU addresses, connectivity and communication, direct or indirect network connectivity status updates (optional), complete load time <15 minutes
satisfying the specific needs of a particular market segment or mission	any system with networked LRUs that have binary loads and reloads, especially for airborne WSs

™ Future Airborne Capability Environment and FACE are trademarks of The Open Group.

Five-Part Product Line Definition: A software product line is …	Definition Applied to the Data Loader CPL
and that are developed from a common set of core assets	architecture requirements, architecture-conforming infrastructure components, standards for loadable software parts, data loader standards, FACE Technical Standard (optional)
in a prescribed way	product line strategy, governance, modeling approach (tool, method, and process)

This definition, the scope definition, and the specification table are all subject to change as the CPL specification modeling team refines the CPL specification. These outputs of Scenario 1A will be baselined for the CPL Organization and delivered to suppliers and other users of the CPLSM. This collection of requirements provides content as spelled out in Scenario 1A for parts of an RFI. This list of requirements is further refined if the CPL Organization decides to proceed with a Data Loader CPL.

4.1.3.3 Use Case Models and Activities

For modeling resulting from the CPL assessment, the elements of a use case diagram (e.g., actors, use cases, extensions) offer insights for the decision process. In the Data Loader example, the use case for the assessment shown in Figure 17: Use Cases from the Data Loader Assessment provides an overview of user-component interactions. The CPL specification modeling team will create multiple use case diagrams that cover a full range of system-context–user interactions along with full activities encompassed by each use case.

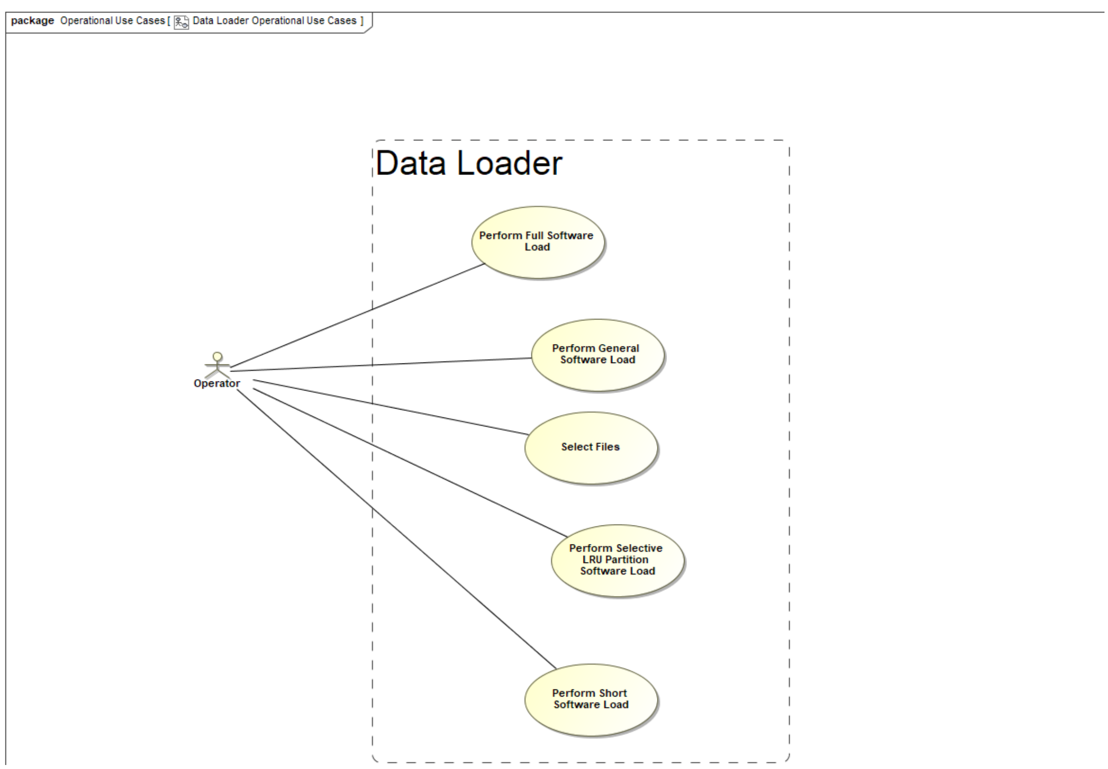


Figure 17: Use Cases from the Data Loader Assessment

4.1.3.4 Feature Concepts Applied to the Data Loader

The assessment described in Scenario 1A for determining feasibility of a CPL approach also results in assessing ways in which instances of a data loader component may differ from one another:

- varying media sources for loadable software parts
- specific configurations of the loader (portable or airborne)
- loader connectivity, LRU connectivity, and protocols (Connectivity and communications are general VPs across all components. The component may include a connection feature set or have options to allow connections to a connection service such as the FACE Transport Service Segment or Transport Protocol Module.)
- alternative connection features that are to be modeled and implemented outside the component
- capabilities for self-test or other operating modes

The assessment of variation will produce informal feature models. During an assessment, these may take the form of a feature table or a FODA feature model as defined in the work of Kang, Cohen, and colleagues [Kang 1990]. The MBPLE feature model is a more formal representation (see Figure 9: A Simple Feature Model in MBPLE). The exemplar covers this representation as part of Scenario 1B in Section 4.3.4: Feature Modeling and Feature Sets and Figure 25: The Model-Based Product Line Engineering of the Data Loader CPL.

The following sections show examples of each form for the data loader.

Feature Table

A feature table is constructed from product data sheets and user manuals. These documents list internal and external features for a base-level data loader product and then list product variants that contain either new or extended VPs. This approach is from the existing product perspective whereas the feature model in the CPLSM is from the requirements perspective.

The features groups cover hardware, software, data, and communication protocols. Within each group, the feature name is identified in the documentation. For this example, product variants are constructed to illustrate the different feature sets that select individual features from each group. The 100% components could be extended by the supplier or integrator to include additional features by using extension and refinement, as shown in Figure 5: Weapon-Specific Instance Model Extended by Weapon System Integrator.

Table 11: Data Loader Product Features and Feature Set Selections

Example Data Loader Feature Group	Core Features	Component Variants (Options)	Component Variants (Alternates)	Feature Set A	Feature Set B
Hardware	Software (SW) Part Media		USB		
			InternalHD	X	
			CD		X
	Network	Connection	429		
			Ethernet	X	X
			AFDX		
			Canbus		
		Deployment	Direct	X	
			Indirect		X
Software	Operator Interactions		SelectMedia		X
			SelectHardware	X	
			SelectPart	X	X
			LoadVerification	X	
	System Functions - FIND - Load direction				
	Find				
	Information				
	Direction		Download	X	
	Reporting				
	Load extent		Short		X
			Full	X	
Qualities	EaseOfUse				
	Cost				
	Weight				
	Performance		<15 minutes	X	
			<10 minutes		X
	Maintainability				

Feature Model and Feature Sets

Figure 18 illustrates the Data Loader features in tree format, according to the feature groups.

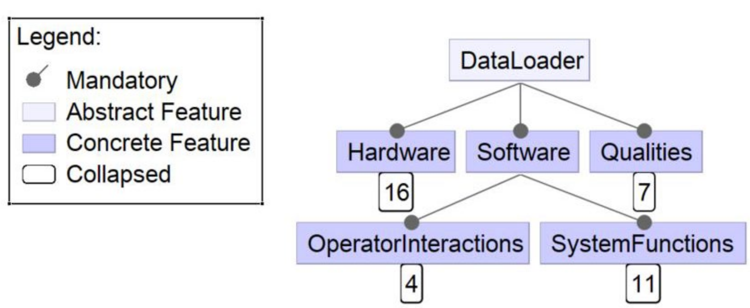


Figure 18: Feature Model Showing Groups of Features

Each group contains detailed features, as shown in Figure 19, for the software function group.

Figure 20 shows a representation of the feature model that captures a selection of features from the feature model. This selection constitutes a feature set for a specific Data Loader instance built from the feature model in Figure 19.

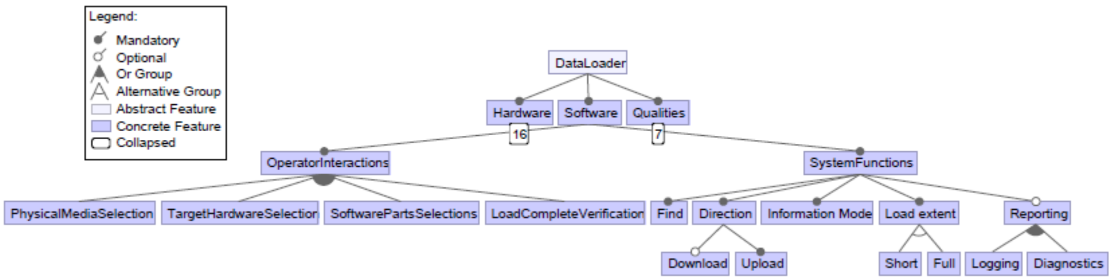


Figure 19: Expansion of the Software Feature Group

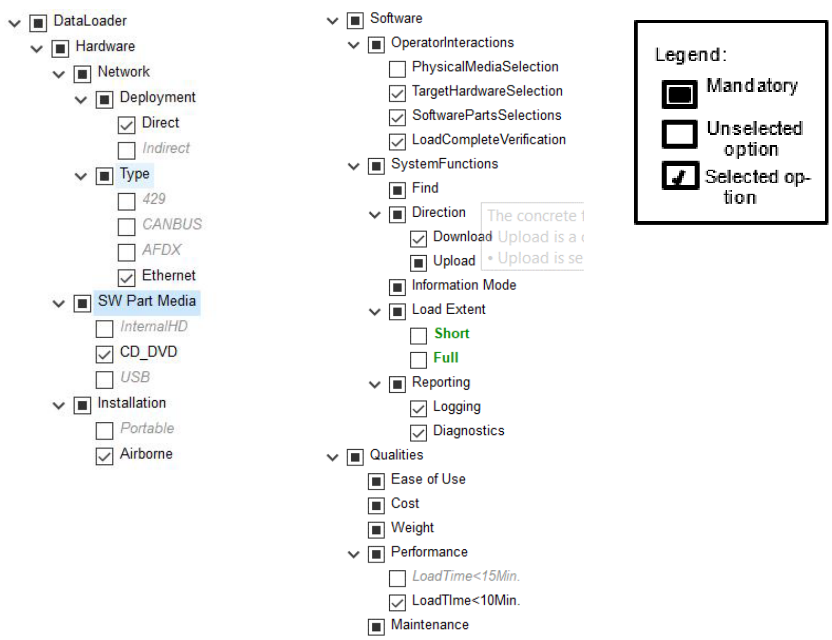


Figure 20: A Feature Set Derived from the Feature Model

4.1.3.5 Quality and Deployment Variations

Beyond satisfying functionality, the CPL assessment may also need to analyze for satisfaction of nonfunctional requirements. Vendors provide data for many of the nonfunctional requirements in the form of specification sheets. They may perform operational benchmark testing of solutions to obtain this data.

The CPL Organization and WS acquirer will assess existing specification sheets, WS needs, and technology assessments. These two groups collaborate to establish specific ranges and properties to propose for a CPL. They also determine means to assess products for the achievement of those specific ranges and properties by components. Table 12 shows an example of the features that must be identified and characterized for a data loader. This representation is a model for specification of the data loader instance.

Table 12: *Marketplace Survey of Nonfunctional Requirements for the Data Loader CPL*

Quality or Nonfunctional Requirement	Acceptable Ranges or Mechanism
Memory size bounds	32–64 GB
Wait times	Latency: <15 minutes to complete entire load Bandwidth requirements and network connection to support: 1–1,000 Mbit/sec Schedulability: satisfy WS cold start <15 minutes
Processor requirements	Portable: not applicable Airborne: <5% of available MIPS
Security	Mechanism: 126–252 encryption
Safety	Mechanism: built-in load status verification
Development needs	Modifiability Extensibility

Under Scenario 1B, this report will expand on the assessment artifacts to address a full example of a CPLSM for a product line of the ARINC-615A Data Loader standard.

4.1.3.6 New Component Offerings

If the CPL Organization and acquirer determine viability for the CPL, they provide the set of artifacts, including the CPL definition, use cases, features, and quality attributes described in Section 3.2: CPL Specification Modeling for the 150% Model and Table 8: Submodels for the 150% CPLSM Artifact. The assessment results form the basis of further modeling by the CPL specification modeling team. The CPL Organization may also provide the information to potential suppliers. These include the built-component suppliers, shown in Figure 21: Supplier Workflow to Apply the CPLSM from Assessment to the Marketplace (right side), who have legacy components that address the informal artifacts. Other potential suppliers may offer development approaches for new components characterized by the component product line supplier (left side) in Figure 21. Either development approach will lead to components that address the given assessment artifacts. A component product line supplier may also propose development or use of a product line production capability that can produce individual components in the product line.

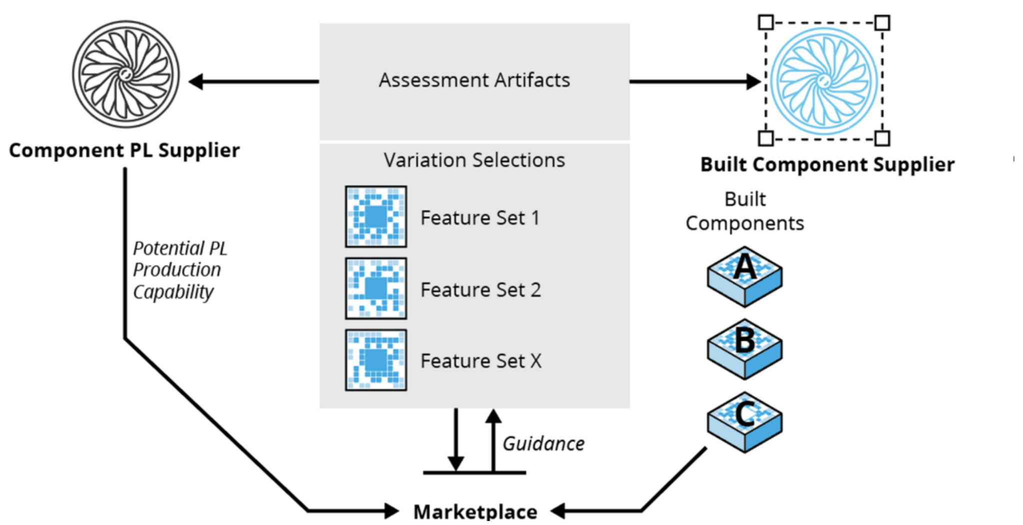


Figure 21: Supplier Workflow to Apply the CPLSM from Assessment to the Marketplace

4.2 Scenario 1B: Create New Component Product Line—Establish 150% Model for the CPL

When the CPL Organization decides on a new CPL, it has the responsibility of establishing the path forward to develop the full CPLSM. In the context of the modeling approach, the CPL specification modeling team builds on the above assessment artifacts to formalize them into a 150% model. In addition, the full activity of creating the CPLSM might involve developing other artifacts such as component simulations—for example, creating instances that can be simulated in a modeling tool. These additional artifacts are for CPL evaluation purposes but are not covered in this report. The CPL Organization evaluates tool, method, and process alternatives to determine the best approach for each CPL. Table 10: Data Loader CPL Definition includes each of these resources as core assets for the product line and incorporates them into the production strategy identified in that table.

Scenario 1A focuses on feasibility: Can the CPL Organization identify potential users for components in a CPL if it is properly scoped and defined? Does the collection of assessment artifacts provide adequate CPL coverage to support the CPL Organization's go/no-go decision to create the full CPLSM? Is the CPL Organization prepared to initiate work with suppliers or other component development teams? Once feasibility is determined and the CPL is properly scoped, Scenario 1B describes the approach for establishing the 150% model. The CPL Organization must build on the assessment artifacts and proceed through a series of model development and evaluation activities. These activities include the CPL specification modeling team and appropriate stakeholders.

4.2.1 From Assessment Models to the 150% Model

This section deals with the need to use established modeling practices and evaluate models throughout the lifecycle. The content of the 150% model includes the refinement of the artifacts enumerated in Section 3.2: CPL Specification Modeling for the 150% Model and Table 8: Submodels for the 150% CPLSM Artifact. These submodels cover the full set of CPL features, functions, data architecture, and other representations. Table 8 provides a complete list with explanations.

The diagram types for representing the content listed in Table 8: Submodels for the 150% CPLSM Artifact may have predefined rules for including specific content in submodels of the 150% model. The CPL specification modelers build these submodels as a collection of materials that includes the graphical representations plus supplementary documentation. Model contents vary depending on the languages and tools used in the representation. Some modeling languages have a single type of representation, either text or graphical, while others such as the Architecture Analysis and Design Language (AADL) have multiple representations such as text, graphics, and XML based. Modeling languages such as AADL and SysML have semantics to represent behavior and relationships of a software system as part of a computing system within the broader WS platform. In some cases, the tools provide extensions to the language standard. These may include tools for analysis, simulation, directives for documentation generation, and often early versions of the next release of the language standard.

"Why," "Who," "What," and "When" are fundamental questions that should be, but all too often are not, asked early and often during refinement of the 150% model:

- "Why is this submodel being created?"
- "Who will use the submodel?"
- "When should the submodel be created?"
- "What other submodels will be derived from content elsewhere in the model chain?"

This section covers the following artifacts for Scenario 1B as established for the Data Loader example:

- Data Loader CPL Modeling Plan
- Data Loader CPL Submodels

4.2.2 Modeling Plan for the 150% Model

The Modeling Plan, introduced in Section 3.2.2: The 150% Model Artifacts, describes the rationale for specific types and quantities of models. This section elaborates on the submodels within that plan. Using the Data Loader CPL as an example, the plan establishes the submodels needed to constitute the 150% model. The plan in this example is not intended to be exhaustive but is representative of the types of submodels the CPLSM team should provide to support both development and evaluation activities. The modeling plan supports the creation of the 150% model. It provides the specification modeling team with guidance in creating the 150% model in terms of its constituent submodels:

- what submodels to produce
- how to use and elaborate the assessment submodels
- how the submodels that the team creates will be used
- what content each submodel should include

MBSE projects create a "model chain," that is, a series of submodels, as the project proceeds through the development process stages. The assessment modeling forms the basis for models created in later phases of the process. The submodels in this first link in the model chain include domain models, use cases, requirements, and others. The submodels created later for the 150% model have traceability and derivation relationships with earlier modeling content. They provide a chain of evidence to boost confidence in predictions of fully identifying feature points, requirements, FA, and behavior to guide design and implementation.

4.2.3 The 150% Model Artifacts

4.2.3.1 Requirements

Requirements may be in table or diagram form and encompass both functional and quality attribute needs and descriptions. This artifact is covered in Section 3.2.3.2: Requirements. For the 150% model, the artifact is expanded to include the relationships between requirements, features, and functions from the FA. The 150% model also includes additional textual information to cover options and alternatives that may be required in a 100% instance model that specializes the 150% model.

4.2.3.2 Use Case Models and Activities

Section 3.2.3.3: Use Case Models and Activities describes the use case content for the 150% model. The modeling plan, to satisfy the needs of the 150% model, must also include component-specific guidance to apply in developing use cases. The scenario identifies one or more potential WS users. These users or communities must be considered in the full range of use cases with VPs. Deployment options, timing constraints, and hardware constraints must be included as possible feature variants in the use cases.

4.2.3.3 Standards

Standards may include documentation, industry standards, organizational standards, or project guides. The level of authority may span the enterprise, product family, or development organization. In Sections 4.1.3: Scenario 1A Example—The Data Loader CPL and 4.3: Scenario 1B Example—The Data Loader CPL 150% Model of this report, the ARINC-615A industry standard is applied to develop the model chain for a CPL of data loaders [ARINC 2007].

An organization may develop a metamodel to specify the models and submodels that the modeling team should create. Figure 23: The Model Chain for the Data Loader serves as an exemplar of the metamodel. It may be applied as the "organizational standard" or "project guide" (see Section 3.2.3.4: Standards) to follow in developing the 150% model as well as the modeling approach for creating individual submodels or reusing specific models.

4.2.3.4 Feature Model

As with other artifacts for this scenario, Section 3.2.3.5: Feature Model covers the representation forms and required information. The scenario considerations include details about the stakeholders who will use the component as well as integrators who will integrate an integration-ready component. The deployment features and VPs they generate will be especially important in that context. Section 3.1.4: Quality and Deployment Variations contains not only deployment but other nonfunctional features that the feature model must include as possible variants.

4.2.3.5 Functional Architecture

VPs to account for specific stakeholders in the scenario drive the specific aspects of extensions to the 150% model. Section 3.2.3: The 150% Model Artifacts provides the basic representation, but the combination of features and WS needs under the scenario will result in supplier or integrator extensions to those features, with corresponding additions to the FA. In evaluating the 150% models, stakeholders will need to assess its extensibility to cover WS specifics in the scenario.

4.2.3.6 Behavior

SysML activity and sequence diagrams are also used to represent component behavior. State diagrams are another representation that can capture behavior or components that have different modes (start-up, operations, error recover, etc.). The scenario captures the interaction between the CPL Organization and WS acquirers. Reviews of the behavior submodels are needed to assure those stakeholders that the 150% model will satisfy not only functional but also system interactions.

4.2.3.7 Data Architecture

Section 3.2.3.8: Data Architecture describes the basis for the 150% model data architecture. Under Scenario 1B, stakeholders provide content from other submodels (e.g., the feature model, use cases, and FA) to elaborate data abstractions. The CPL specification model team uses these abstractions to satisfy the potential WS users of components in the CPL. To satisfy these users, the data architecture also captures variations of the data abstractions via VPs. The data architecture must also support interoperable means of data exchange among components.

The Open Universal Domain Description Language (Open UDDL) provides the rules that guide the conceptual basis for detailed data definition and data logic of the abstractions of the data architecture [Open Group 2023]. The data models help represent specific data constructs that are required and the format used for common application across a single or multiple CPLs. Open UDDL is the basis for data design that applies the CPL 150% data architecture. Modelers select the data concepts and logical abstractions to define data structures and elements within those structures and then to define flows within or between components. A platform data model uses specific data types for component implementation and integration and for exchanges across the weapon system.

4.2.3.8 Architecture Requirements

Stakeholders in Scenario 1B will need to review the architecture requirements in the CPL 150% model for consistency with their own understanding and the needs of their WS. The individual WS may have additional architecture requirements that will be specified for it. The evaluation of potential CPL candidates for the WS will determine if the CPL satisfies those requirements, can satisfy them with weapon-specific features, or will require the integrator to perform further tailoring. This review may help the acquirer determine that the CPL is not adequate for the system.

4.3 Scenario 1B Example—The Data Loader CPL 150% Model

Under Scenario 1B, the CPLSM develops the full 150% model for the Data Loader CPL. The assessment artifacts, with the addition of several other submodels, provide much of the content for the 150% model. In addition, the system architecture requirement for the 150% model is that a single-tool approach be used for the model chain, to the extent possible. For this example, the CAMEO® System Architecture tool can represent all submodels of the 150% model by including the CAMEO plugin for MBPLE [MBPLE 2021]. While the tool satisfies that requirement, a CAMEO-only solution may not be the optimal approach to satisfying the requirement. Plugins that are alternatives to those from No Magic, such as pure::variants® from pure-systems or Gears® from BigLever, should also be considered to satisfy the requirement.

4.3.1 Context Diagram

ARINC-615A specifies a device that takes loadable software parts—software executables defined by ARINC-665—from a storage medium and delivers those executables to target hardware on a network [ARINC 2007, 2019]. The target hardware may be individual LRUs or other devices. The ARINC-615A standard defines an ethernet protocol by which the data load application (DLA)

® CAMEO is a registered trademark of Dassault Systèmes. pure::variants is a registered trademark of pure-systems GmbH. Gears is a registered trademark of BigLever Software, Inc.

communicates across the network to LRUs to deliver software parts. However, application of this protocol should be considered a deployment variation. Bus architectures other than Ethernet are one variation. To support interoperability system architecture requirements for using other NDI components, a deployment variation may apply a different component scope. This approach creates a feature set for components that communicate with transport services independently of a particular network architecture or protocol. The full feature model must consider this degree of variation.

Figure 22 provides the SV-1 DoDAF model or Systems Interface Description for the data loader (shown in the upper right of the diagram). The loader communicates via the network, here identified as compliant with the IEEE 802.3 Standard for Ethernet, to target devices. Variations may include

- loader types: portable, installed
- network: the type of network that supports the data loader to target loading
- protocols: the protocols for communicating across the network to affect the load
- loadable software part storage (not shown in this view): mass storage device, thumb drive, built into loader, and other types
- deployment: to allow for variations in connections through a transport service to a virtual network

The example covers models and other representations related to the data loader as they exist in a model chain. The purpose of the example is to show information captured by each representation, model, or submodel. Information in each representation is used by other models, as described in Section 3: Creation of Models in a Model Chain, across the model chain.

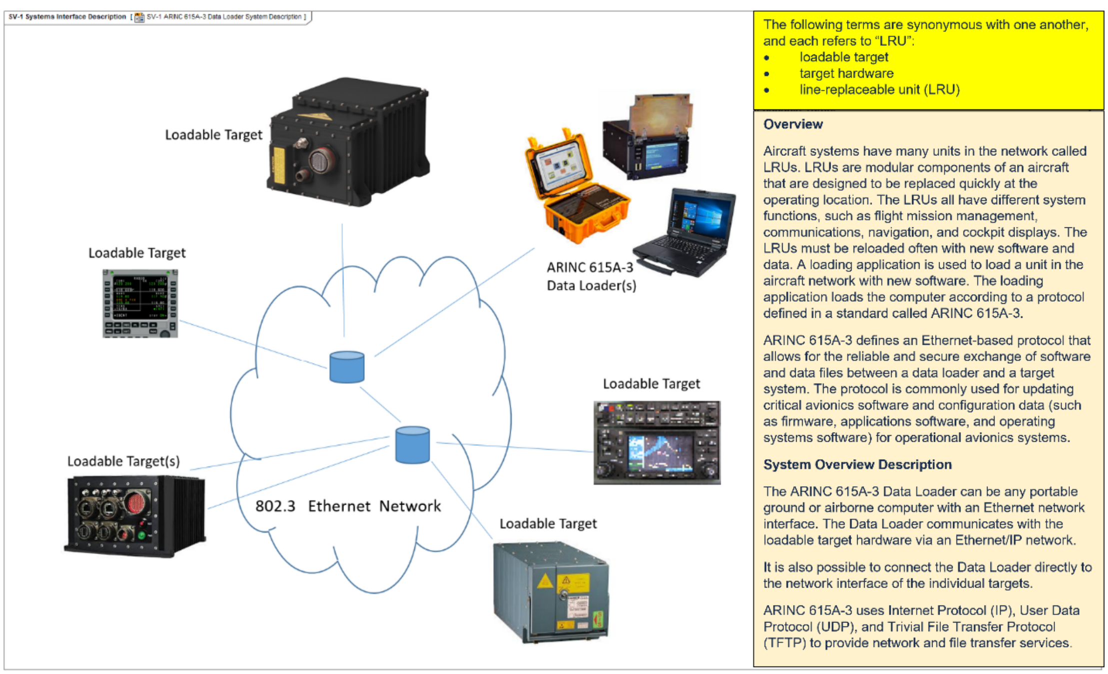


Figure 22: Operational Concept of the ARINC-615A Data Loader

4.3.2 Example Overview

This section of the report describes development of example models for a CPL that satisfies the ARINC-615A Data Loader standard. Each model supports further development and delivery of a product line of data loader components by component suppliers as part of the overall technical data package for inclusion in a product line marketplace [Cohen 2022]. Models are created during the development. These models satisfy the following model chain needs to identify and provide understanding of core assets:

- requirement identification and analysis
- application of domain expertise to define features and feature sets
- descriptions of existing architecture artifacts or models used to create the initial architecture description
- support for architecture evaluation
- design of internal model information used by implementers

The ARINC-615A Data Loader example illustrates activities that contribute to creating the model chain for the component and for capturing variation across the product line. It is exemplary only, for the purpose of identifying and illustrating model and submodel types using a variety of modeling languages. The example represents only a small part of the complete activity to deliver a CPL model that could be used to implement actual data loaders from the marketplace. Figure 23: The Model Chain for the Data Loader represents the model chain for the data loader component. Each area of this model chain is covered by the example. The individual 4+1 views (use cases, logical, physical, process, and development) are represented in the figure as subtypes of the type “4+1 views” [Kruchten 1995]. These elements are modeled by the component supplier to address the CPLSM. The views include structure, interfaces, and design elements needed to support both component analysis and implementation. This report does not discuss those aspects of the 150% model.

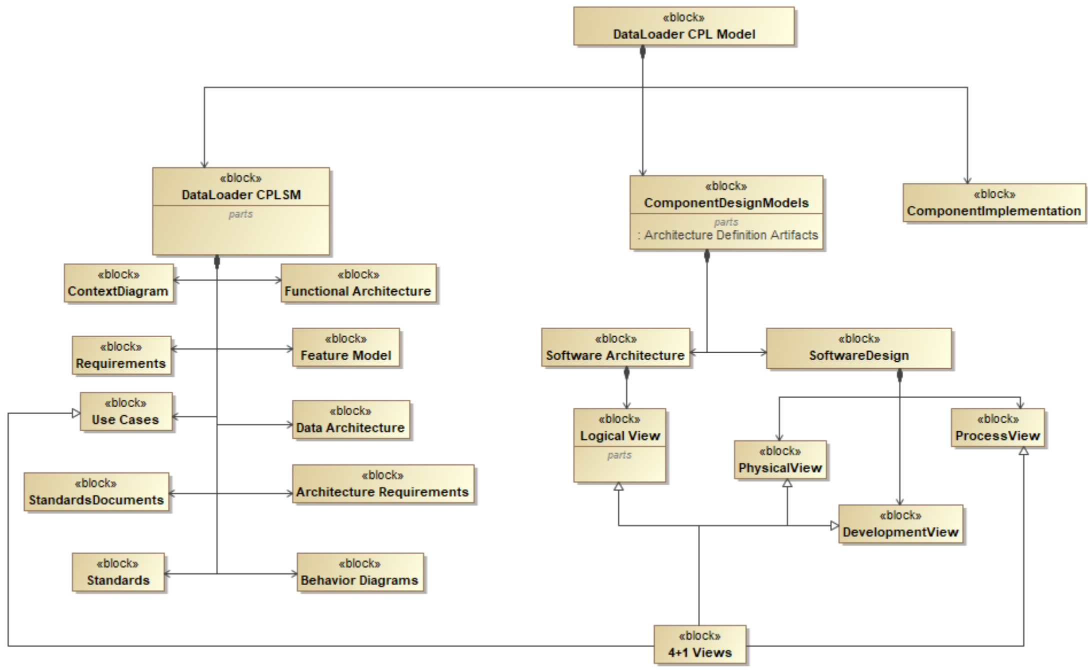


Figure 23: The Model Chain for the Data Loader

The left side of Figure 23 shows that submodels of the model chain are used by the acquirer and WS integrator to make selection decisions about the data loader component. Component variations are considered as options for acquisition and integration. The variations are collected as feature sets to define individual data loaders, but they are independent of the WS that will use the component and are created without knowledge of WS specifics. The goal of the Data Loader CPLSM is to be general enough to guide the supplier across the model chain to define the 100% instance model. That model will form part of the WS specification and will define the requirements for suppliers of data loader components or of a component production capability.

Developing a single model that can be evaluated from different views provides insight into features and VPs associated with each view. It also provides insight and support to collaborate among different stakeholders of the product line, such as the WS integrator, WS acquirer, component specification team, and component supplier. All views are relevant to each of these stakeholders.

The component modeling for the Data Loader CPL in this report is based on the CPLSM descriptions that comprises government-furnished information (GFI) for a data loader component product line. These apply the ARINC-615A standard for the data loader and in general identify functionality and decompose functions. Behavior submodels are needed for component specification; at the component level, they must document the externally visible states and transitions that the component should exhibit. Variations of a base component can include introducing and removing states. Variation in behavior can arise from features that may be captured as deployment, functional, or data features. These variations may also be used with component interface features.

The following sections describe information mining, data modeling, and associated views of the component, its features, and VPs captured in the model:

- **How much detail should models show?** Most of the operations found in the scenario description can be found in the SV-4 functional decomposition. However, an SV-4 may contain detail needed for design and implementation but not for views that emphasize the externally visible features of the data loader. The SV-4 should also include VPs. A VP on a function represents the choice of options or alternatives in the function.
- **How should a logical view be presented?** The logical view shows the SV-4 information in a structure form of a block diagram with data flows. It is at a high level and does not represent design information; a block does not represent a design module, only the SV-4 functional hierarchy grouped to identify data exchanges based on the information model.
- **What level of process modeling does the data loader require?** The SV-10c captures behavior in a sequence diagram. Additional iterations of behavior modeling can flesh out the behavior needed to arrive at an appropriate level of detail.

4.3.3 Sources for Specification of the Data Loader

4.3.3.1 Information Mining and Scenario Development

This Data Loader CPL example draws on information sources that are most frequently used by suppliers of ARINC-615A Data Loader standard components [ARINC 2007]. These include

- data contained in a GFI Cameo model developed by a CPL specification modeling team
- manufacturer's product data sheet descriptions of typical equipment that can become part of a product line
- ARINC or other industry standards that may be applicable to the design of the equipment
- standards that may be relevant to the context that the equipment must operate within and communicate with

For the Data Loader example, the GFI Cameo model included

- use cases
- SV-1 Systems Interface Description: describes the composition and interaction of systems, incorporating the human elements as types of performers (organizations and personnel types)
- OV-1 High-Level Operational Concept Graphic: describes the mission, class of mission, or scenario, showing the main operational concepts and interactions between the subject architecture and its environment and between the architecture and external systems
- SV-4 Systems Functionality Description
- OV-5A Operational Activity Decomposition Tree
- SV-10b Systems State Transition Description
- SV-10c Systems Event-Trace Description (sequence diagrams)
- feature model of required and optional features of components in the product line (part of this example but not in the GFI)
- functional and system architecture requirements

4.3.3.2 Use Cases and Scenario Development

Figure 24: Use Cases from the Data Loader CPLSM shows a fragment of the scenario for the use case "Perform Full Software Load." (See Appendix A for the complete use case scenario description and how its content can be mapped to elements that the different views should reflect.) For example, the "Perform Full Software Load" use case may reflect one form of variation in the software to support short loads, as well. This element of the use case can also be a source of variation in deployment to hardware. The scenario may describe alternative means of connecting to a network where some components in the Data Loader CPL include a network interface protocol and others rely on network services to perform loads. The use case "Select Load Direction" refers to an operation to send parts to a destination or receive information back. The use case variation exists since download from a destination is an optional feature. A use case for monitoring upload status (not shown) periodically receives logging data to indicate the status of the process. It is an option that assumes the component will include the network interface as well as receive status updates. An alternative component may not wait for status feedback in a load exchange.

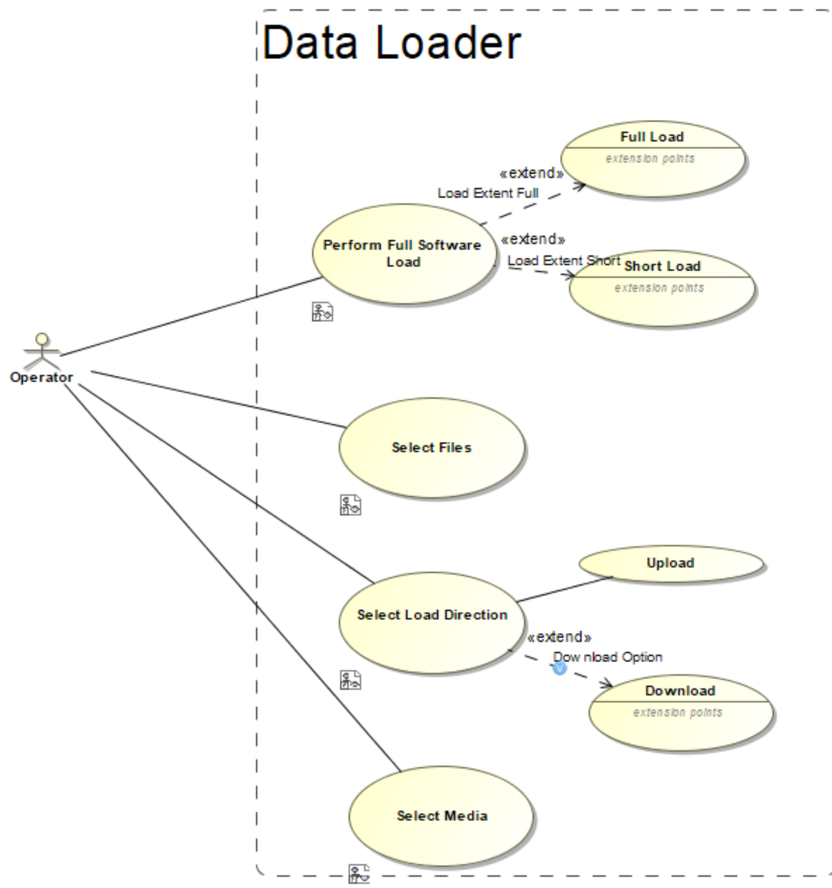


Figure 24: Use Cases from the Data Loader CPLSM

4.3.4 Feature Modeling and Feature Sets

The Data Loader example demonstrates each category of features for defining the CPLSM, as shown in Table 13.

Table 13: Feature Groups Applied to the Data Loader

Category	Application to the Data Loader CPLSM (examples)
Core	Core features include the following: obtain loadable software parts from storage, identify software part destination and binary, and format binary distribution to target LRU
150%	Storage media variants, download from target, data formatting for logging, timeout sensing, portable or onboard operations, connectivity options
Instance Model	Example: core plus hard drive as medium, no download capability, format data for logging, timeout sensing, portable
Feature Set	Collection of features of potential instance models, including established physical components from suppliers and the most likely acquirer needs based on assessment results
Integration Features	Handle non-ARINC-615A standard loadable parts

Figure 25: The Model-Based Product Line Engineering of the Data Loader CPL illustrates the MBPLE approach to the feature model as built on concepts presented in Section 3.2.3.5: Feature Model. The example shows much of the same content as in the assessment feature model, but it applies the MBPLE profile. The main aspect of this model is the combination of mandatory features that form the core features for the Data Loader CPL and the extensions of those features through options and alternatives. The CPLSM modeling team is responsible for not only identifying the nonmandatory features but also supporting creation of feature sets that exploit nonmandatory features to meet the needs of individual WS acquirers. The full range of features also provides requirements to the CPL component suppliers, either third party or integrators. The ability to satisfy the widest range of feature sets and feature specializations is a primary differentiator in the acquisition of components and is a key element in Governance Scenarios 2 and 3.

Feature sets are the other primary application for the feature model. The MBPLE feature model supports the formation of feature sets through the instance tool. See Figure 26: Feature Selection for a Feature Set. This tool enables two steps:

1. Parts selection: Any potential WS acquirer wishing to evaluate and select components from the CPL begins with the instance creation to determine if the feature model and feature selections are adequate to address its needs.
2. Diagram or instance creation: The resulting diagram can be examined and further analyzed to understand relationships across features as a specialization of the 100% model version of the 150% model.

Product data sheets and product descriptions on vendors' websites are used to identify typical hardware devices, electrical devices, and communication interfaces. The data sheets also cite device standards and physical, electrical, and communication properties that could be attached to the modeled components for use in analysis of the data loader itself but also in a larger system setting, such as connecting the data loader to various LRUs. Modelers should also note that component specifications apply to components within an existing product or to the product itself within a larger system-of-systems setting. Both contexts will have the same characteristic categories (e.g., interface, behavioral), but they may have different types and amounts of data.

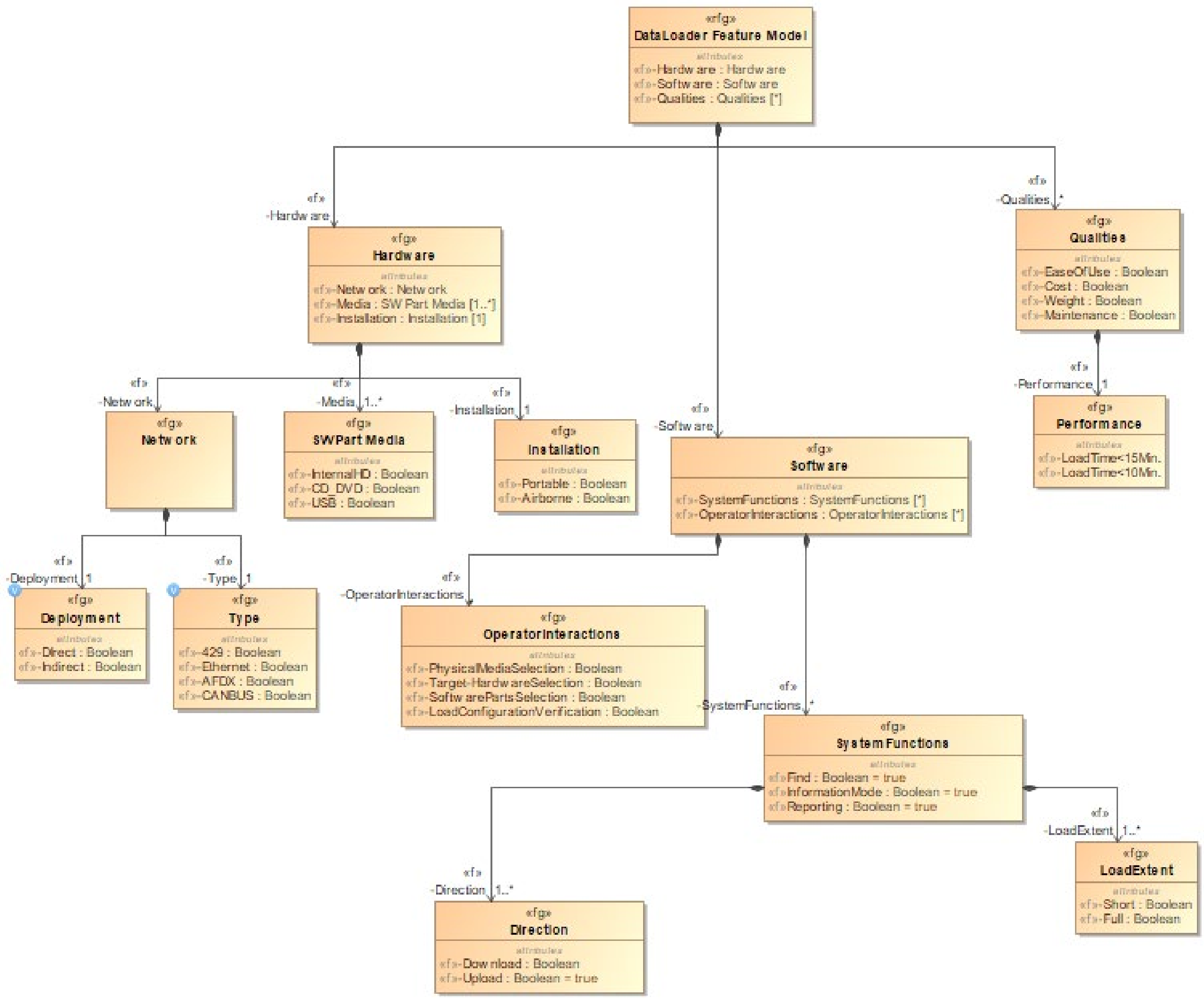


Figure 25: The Model-Based Product Line Engineering of the Data Loader CPL

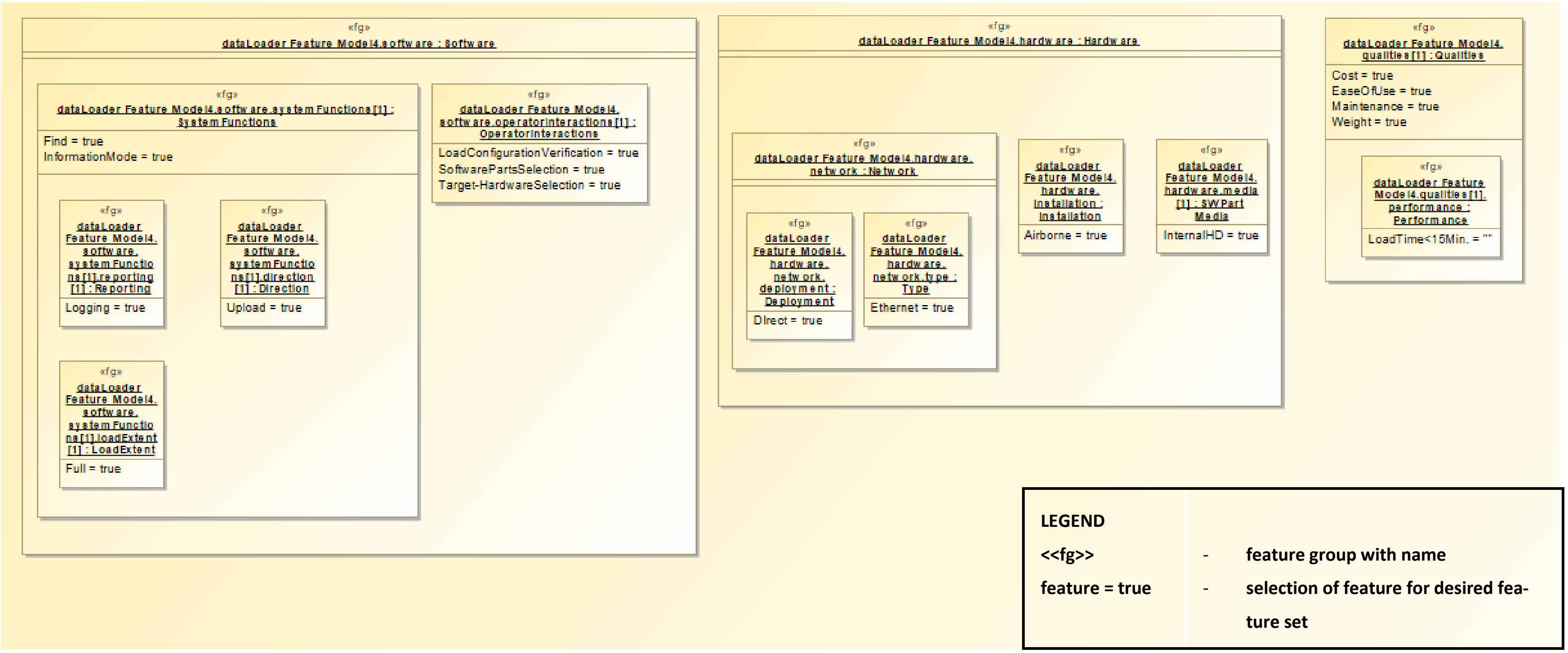


Figure 26: Feature Selection for a Feature Set

4.3.5 Functional Architecture

Component decomposition leads to a set of key component abstractions, highlighted in SV-4 and SysML representations. This decomposition supports functional analysis but also helps identify common and varying elements across the components in single or multiple views. For a top-level abstraction of a component, the decomposition should include a behavioral description of its operation that identifies its operational states. Furthermore, behavioral descriptions of interface features specify valid states of the interface.

4.3.5.1 Determining the Components and Appropriate Level of Abstraction

The following data sources were used to generate the FA:

- Sequence diagrams helped identify functional elements and data communicated to other elements and allowed derivation of connections among elements. Connections among multiple elements provided the basis for constructing and assessing end-to-end flows through the component.
- The SV-4 provided insight into the data flow and functionality of data loader software. For this example, product data sheets provided externally visible features that the software, hardware, or both would have to support. Product data sheets also provided performance and physical properties that could be annotated to the model and used in analyses, such as for power or weight.
- The ARINC-615A standard contained specification data, descriptions of sequential operation, operational behavior, and states.
- Use cases, composed of a number of scenarios, contained information about optional operational features and diagnostic capability.
- User manuals and concept-of-operation documents provided use case descriptions such as data uploads and error types. Finer grained description could be viewed as scenarios, such as types of uploads and user response to errors.

4.3.5.2 Modeling with DoDAF Viewpoints

The DoDAF models under the Systems, Service, or Operational Viewpoints provide the FA and behavior for the CPL [DoDAF 2010]. In this section, alternative representations use the Unified Profile for DoDAF in Cameo, where the modular structure can be captured with the SV-4 Systems Functionality Description diagram. The tree structure of this diagram can be used to show the decomposition of the component into functions and to show the decomposition of functions themselves if necessary. It also includes the VPs that are part of the FA for the 150% model. However, it does not show runtime behavior, data flow, or interactions among the different entities. Figure 27 shows this diagram for the functional decomposition of the DLA. The level of detail is consistent with the level of detail in the use case description. However, some types of analyses may need more detail. For this application, there is a standard, ARINC-615A, that describes how these functions must work. Therefore, there is no need to model with more detail and replicate what the standard already specifies unless that further detail will support analyses or virtual integration.

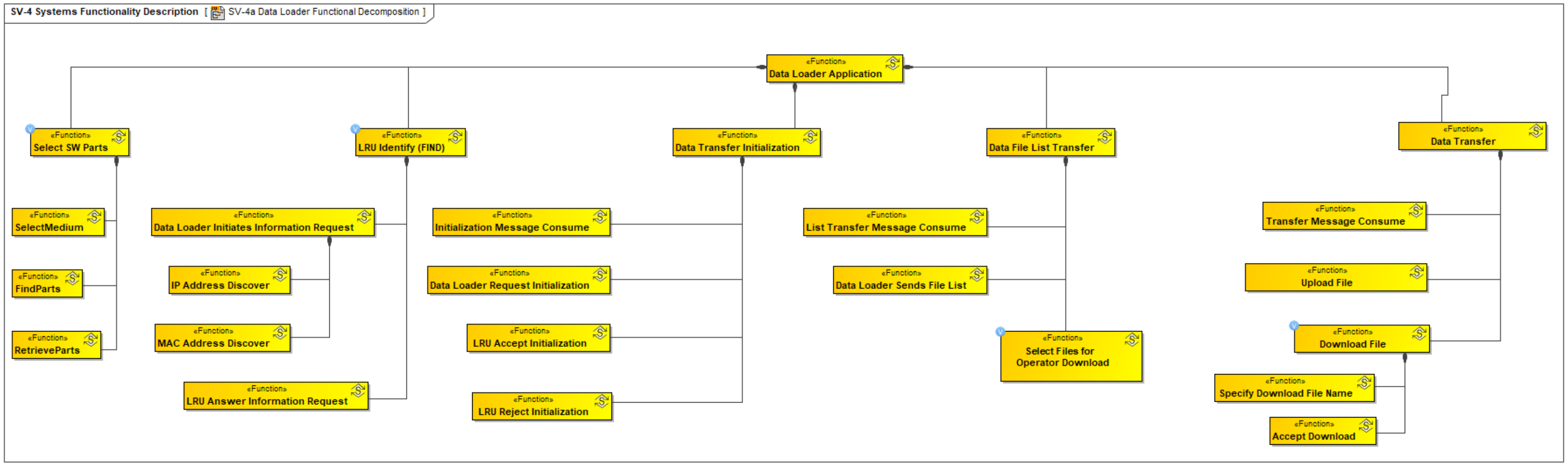


Figure 27: Use of SV-4 to Represent Functional Architecture

Figure 27 also shows variability in the Logging function. Session Logging is required, but Protocol Logging is an optional function. One limitation of using the SV-4 Systems Functionality Description diagram is that, by itself, it cannot represent behavior in the form of control and data flow. To achieve that, it must be combined with the SV-4 Systems Functionality Flow Description diagram, an example of which is shown in Figure 28.

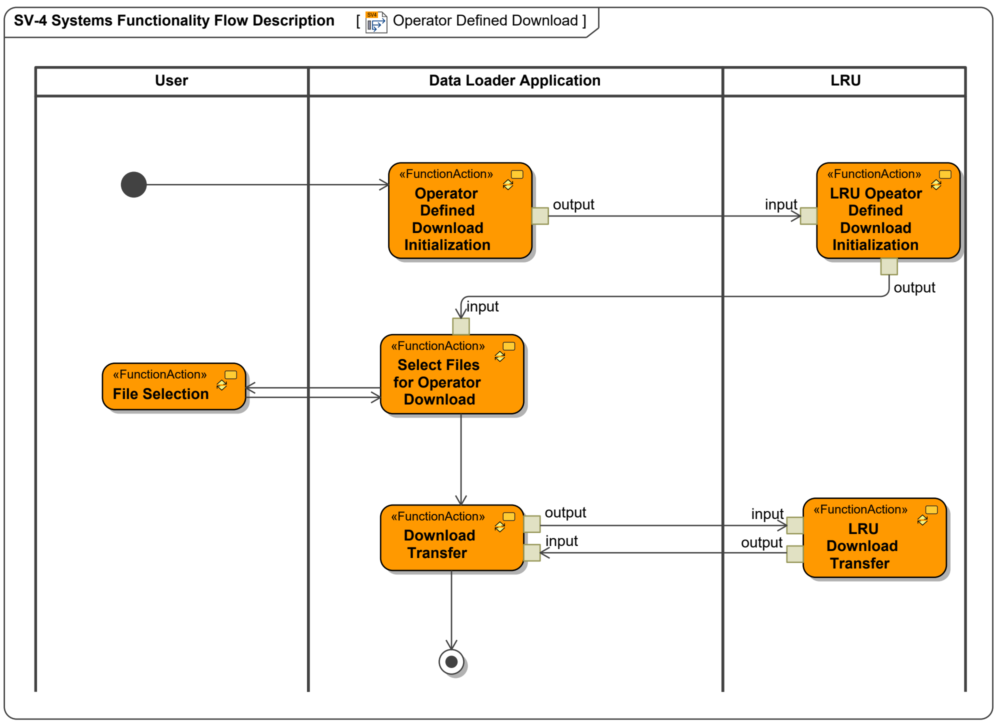


Figure 28: SVC-4 with Flow Description to Represent Logical View

4.3.5.3 Potential Analysis of Abstracted Model

Analysis of the FA representations can be useful in several ways. Generally, the WS acquirer or integrator will begin by applying the CPLSM in the WS specification model. If desired, the WS integrator may apply the component models to consider integration in the context of the WS: Does the model cover the necessary characteristics and properties for the needed WS component? The specification modeling team provides feedback to the supplier and integrator on component properties and use.

Analysis performed at the WS level can also verify that overall system requirements can still be met by incorporating the new proposed component. The component specifier can use the analysis of the WS integrator who incorporates the CPLSM to verify that the architecture of the WS meets its requirements with a properly designed and implemented component from the supplier. The component supplier can use the constraints from this CPLSM to refine the model development, using analysis of the internal design of the component to ensure that it meets the constraints of the integrated WS.

Modeling at this level provides the basis for certain types of analyses and consistency checking. Specifically for the FA of the data loader, model analysis can confirm model consistency across submodels—use cases, features, and data. Flow paths indicate logical flow of information, and critical flows can be defined within the component (Table 12: Marketplace Survey of Nonfunctional Requirements for the Data Loader CPL). Beyond the content of the 150% CPLSM, suppliers will support variations in the performance information. Flows can be annotated with variations in expected timing information, which provides insight about potential latency to the acquirer or integrator. Timing values are budgets associated with a component within the CPL, and the acquirer or WS integrator should use them to ensure that system requirements for things like total data transfer time can be met when the component is used. The data from the CPLSM team is a rough estimate but can be used for evaluation purposes to consider selections among alternative feature sets or supplier models.

Components as well as systems can be active in certain operational modes. Flow analysis in the context of modal operation can provide verification of what components are active in certain modes as well as timing of end-to-end signals within a set of system modes. The SysML models of the data loader apply this collection of information. In SysML, the model for the FA includes both a block diagram of the data (Figure 29) and structural submodels (Figure 30), where the structural model applies data flows parallel to the SvcV-4.

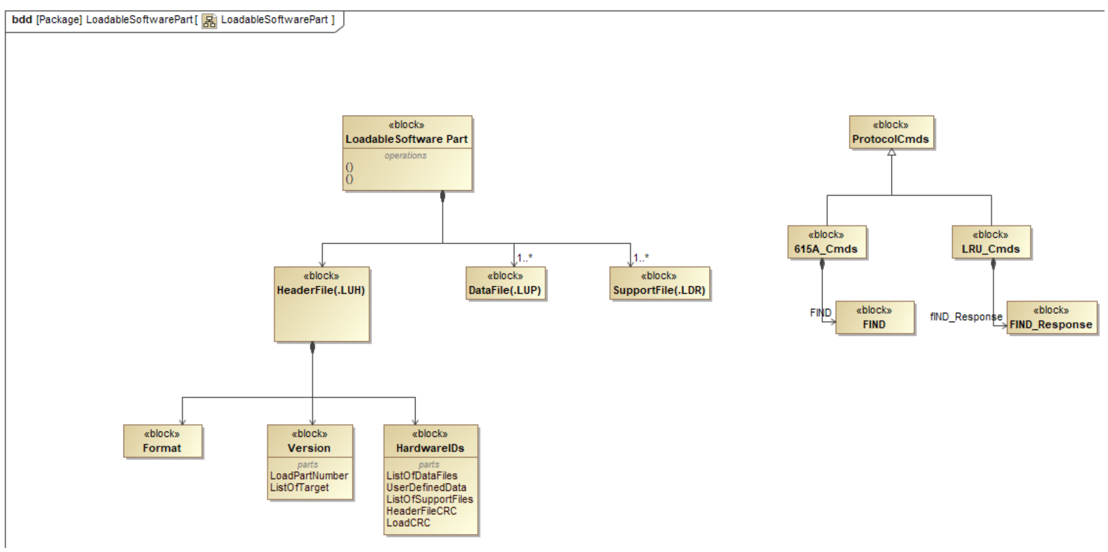


Figure 29: Data Submodel for the Data Loader (Partial)

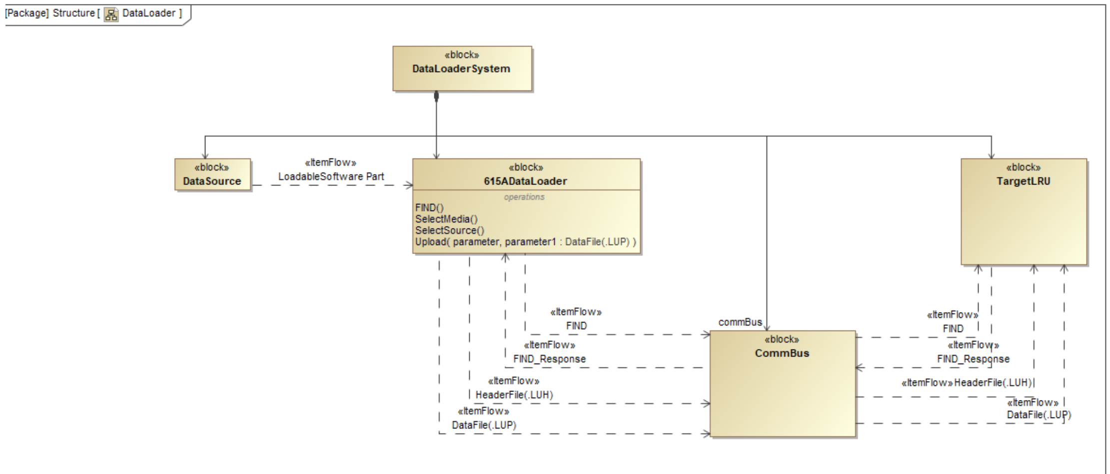


Figure 30: Structural Submodel for the Data Loader (Partial)

The SysML model highlights data exchanges (e.g., LRU Identify [FIND] command and FIND response) as key abstractions of the logical view. These data elements are identified from the SV-4 and contents of the ARINC-615A standard and provide exchange from the data loader to LRUs. The data loader communicates with the DataSource to obtain the software part containing a software data file to be loaded and its destination LRU.

4.3.6 Representations for Behavior

The sequence and state diagram are optional submodels to show functional elements that contribute to understanding component behavior. The example is limited to the highest level and must be expanded to support internals that map elements to functions and data items in the logical view.

Figure 31: SV-10B State Model for the Data Loader CPL is a DoDAF model that shows the information exchanges to support the states and state transitions. The diagram shows the data loader in the full context of the parts storage medium, the network, and the LRUs. The user initiates the upload process by activating the loader (Start DLA). The loader then performs the steps in the state diagram to find target LRUs and to perform the upload software part operation, looping through until all parts are loaded. For this example, the process view assumes successful processing throughout. A complete process view must also include error handling for such hazards as timeout and incorrect checksum on a transmission. The state diagram will show these error states, and the sequence diagram will provide the feedback elements in the loop to account for error handling and correction.

State transitions are depicted in the parallel SysML diagram in Figure 32: SysML State Transition Equivalent of Figure 31.

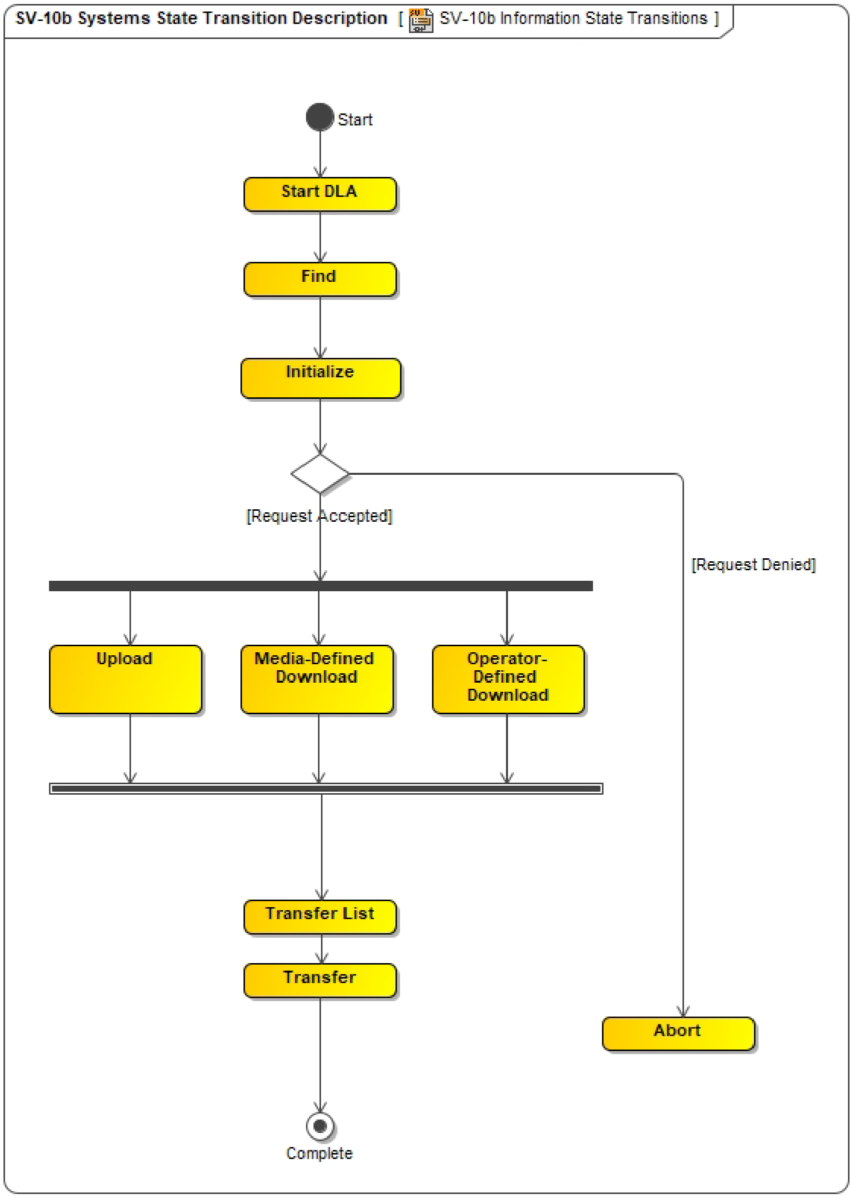


Figure 31: SV-10B State Model for the Data Loader CPL

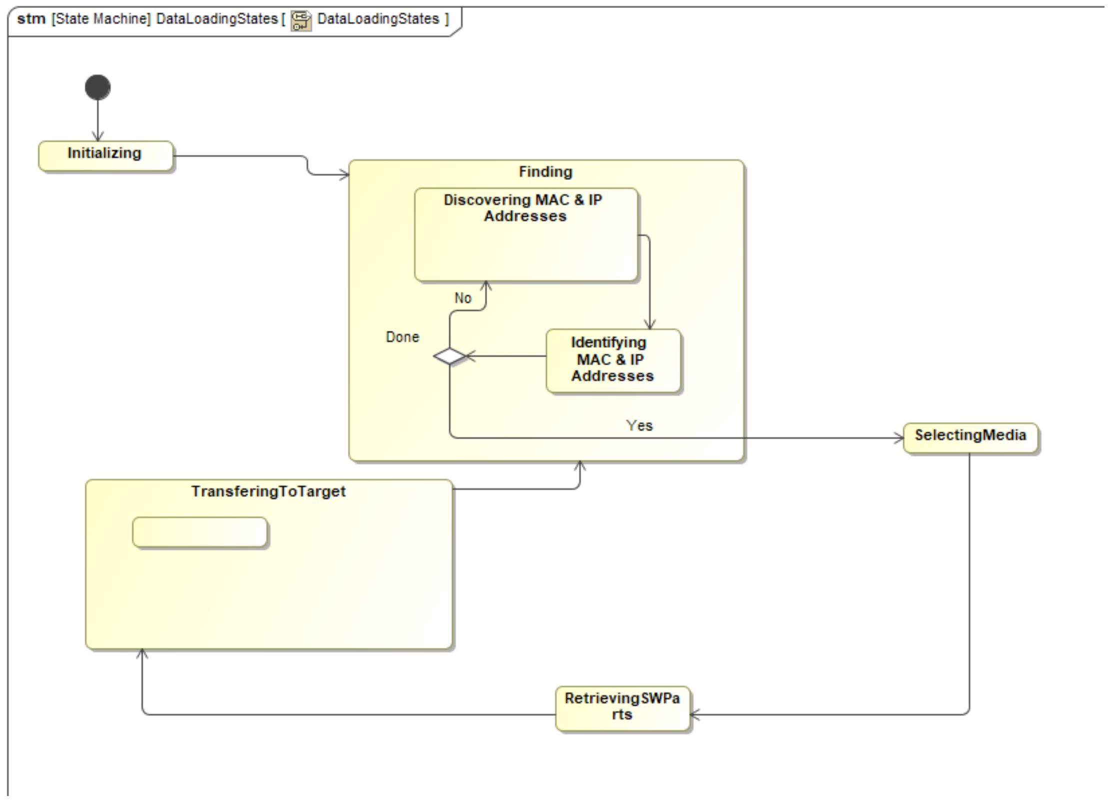


Figure 32: SysML State Transition Equivalent of Figure 31

Event trace models in the DoDAF SV-10C show interactions between component elements from the SV-4 [DoDAF 2010]. These interactions are highlighted with control and information flow between elements. The VP for deployment could be applied here to show that data loader interactions with a network are mitigated by a transport service. This variation is needed to enforce a separation of the component from knowledge about its use by a WS—some WSs will integrate a data loader to the network. This will always be the case in the portable feature. But for installed components, the data loader has options for how it can be integrated. The SV-10C corresponds to the SysML sequence diagram in Figure 33: SV-10C for Data Loader Sequencing. A VP to address deployment features for network connectivity (direct or indirect) will enable alternative timelines to model the direct connection to the network of connection via a network service.

Figure 34: Timeline in Sequence Diagram Showing the Upload Loop shows the state and state transitions of the data loader component using the SysML state transition diagram. After initialization, it

1. finds target LRUs
2. selects software parts from the storage medium for uploading
3. retrieves those parts and processes the content of the part. Content determines where a part should be sent and the data portion of the part to be sent.
4. transfers the part to the LRU

After Step 1, the data loader enters a loop to perform each upload.

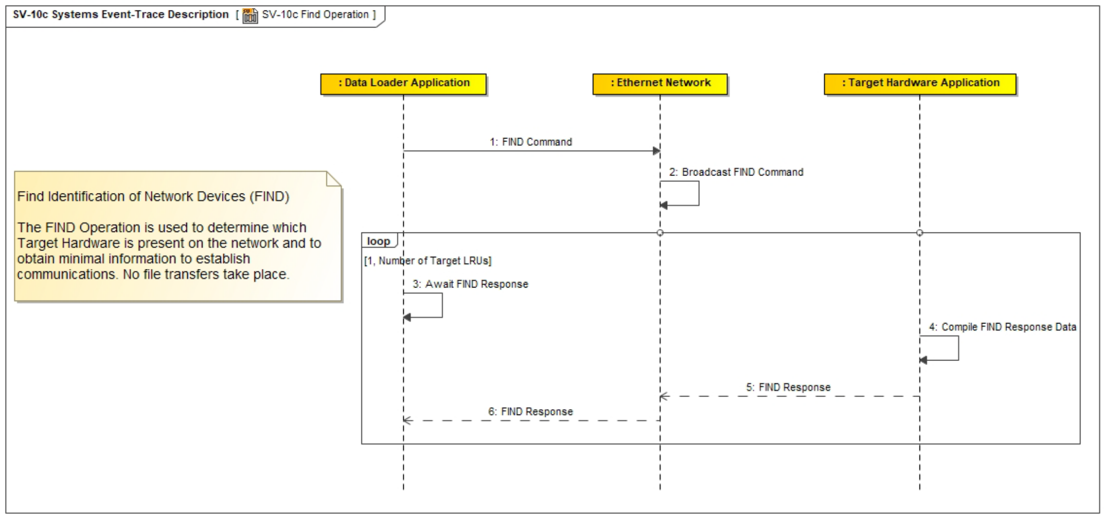


Figure 33: SV-10C for Data Loader Sequencing

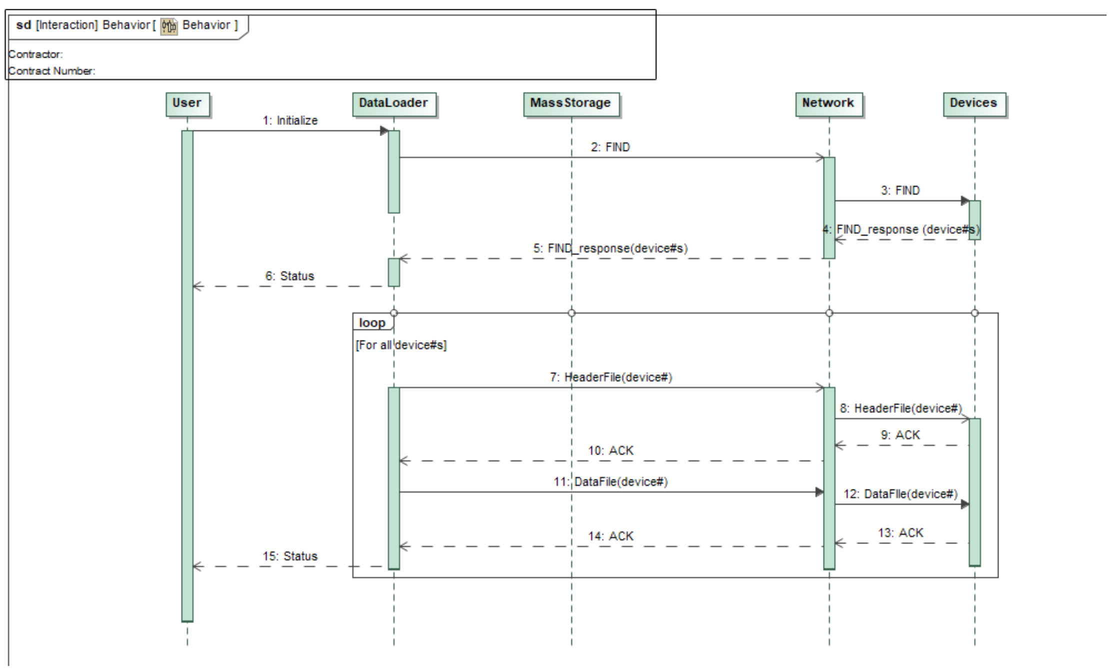


Figure 34: Timeline in Sequence Diagram Showing the Upload Loop

4.4 Scenario 2: Modify/Update an Existing Component Product Line

This scenario has the following preconditions:

- Several components from a CPL are in use in specific WSs.
- A new WS acquirer would like to use Component X from the product line in its new system, but the requirement includes a new data element that would change one of the feature sets of Component X, and two WSs (Platforms A and B) use that feature set.

- This new data element requires a change to design that implements a core feature of the CPL and the interface between Component X and the WS.
- Platforms A and B do not yet have the requirement to update the feature set to their systems.

The governing board must choose between alternative courses of action:

- Accept the change to the feature set for Component X and modify the component.
- Identify a new VP within Component X such that use of the new data element is an option and a new variant of Component X can coexist within the CPL.
- Reject the change, forcing the new WS to either use the existing Component X and supply the new data element on a new interface or take some other approach to incorporate the new data element.

As current consumers of Component X, Platforms A and B must be part of the decision-making process. Several factors could sway the governance of this decision:

- Platforms A and B may experience pressure to adopt the updated feature set, anticipating an upcoming change to their requirements.
- An upcoming unrelated change to Component X could necessitate a new version anyway.
- Platforms A and B may be approaching their end of life, and the timing might be right to retire the legacy component and replace it with the new Component X.

The product line strategy is built on a collaborative approach among the various roles to model, produce, refine, and use components. While the initial 150% feature model for a CPL seems correct to the specification modeling team during development, understanding of what is really needed for implementation will change. From specification to creation of models of architecture artifacts and design by suppliers, the modeling team will capture the "as-designed" components needed by the acquirer or integrator as CPL model instances. Also, suppliers will have components that existed before the availability of the 150% model and specific feature sets. The suppliers will align these preexisting components to the feature sets but may not offer a 100% instance that covers any complete feature set.

This feature set coverage constitutes the component suppliers' view and may include new VPs as well as features that differ from the original feature set or sets. In this scenario, that supposed need for changes in fundamental aspects of the CPLSM actually reflects back on the CPLSM team as an architecture requirement: Investigate areas likely to change, and capture those as VPs in the 150% model. Properly specified, VPs, multiple feature sets, updates to a feature set, or creating new feature sets can lead to new components within an existing CPL without affecting others. The CPL specification should address the question "Does the specification support changes within a product line that will not affect existing components?" This flexibility within the CPLSM must also address the potential cost of extensibility against the loss from not addressing potential users.

Many issues arise as the CPLSM evolves during development of the 150% model. Decisions about the component scope and features may result in derivations from the specification. These derivations come in various forms:

- derived requirements, where an existing component has filled in gaps in a specification
- assumptions, when the specification is vague or open to interpretation

- application of expert knowledge from similar products that designers or implementers bring to the development
- identification of a “component framework” that allows a CPLSM team to identify other models that can be used at VPs

Based on these decisions, the CPL and potential instances will better reflect the stakeholders’ initial intent or understanding. For example, during a review, a stakeholder in the development of the CPLSM may see the architecture requirement for identifying and modeling for change. This places responsibility on the CPLSM team to provide specification of how the component instances should be built in terms of standards, best engineering practice, and stakeholder directives. By applying need for change as a means of identifying VPs, the team creates a 150% model that will be less likely to require changes to the CPLSM. The stakeholders will ask questions like these:

- What industry standards are applied, and how are they documented?
- How does the architect, designer, or implementer know that quality attributes are addressed and achieved?
- Do features provide adequate coverage for the potential range of systems that will use component instances, and do those instances capture the desired feature sets?
- What if the development uncovers the need for new features and features sets? Can existing or new component instances address those enhancements?
- Does the design properly integrate the provided artifacts to achieve appropriate reuse requirements?
- Is the specified data architecture realized throughout the product stages—data requirements through data implementation and specified architecture connections?

The WS acquirer or integrator must evaluate, select, instantiate, and integrate a specific component in a WS. The integrator is responsible for integrating the components into the physical implementation as computing elements of the complete WS or as elements that can be incrementally integrated. Capturing variations as areas for change requires the acquirer or integrator to provide a greater degree of specialization within the 150% model, but capturing VPs at any level within that model avoids late discovery that no feature set adequately addresses WS needs. This approach also avoids the inability to instantiate a component that the acquirer can fully integrate and use. It allows sustained use of the CPL through reinstantiation from the CPL using an alternate feature set or feature selection at VP.

For this scenario and Scenario 3, the logging and diagnostics features offer an example of product line modification or update. The feature model identifies these as optional features within the SystemFunctions feature group. The existing components satisfy a feature set that includes both optional features, but both logging and diagnostics are built into the component. WS A and WS B are using this component.

The CPL Organization has determined a need to build separate CPLs—one for logging and another for health management. The Logging CPL can support the logging feature of the Data Loader CPL, and the Health Management CPL can support the diagnostics. A new WS, WS C, would choose the integrated set of CPLs that support the logging and diagnostics features. WS C can apply both the Logging and Health Management CPLs to other components that it will need.

The CPL Organization determines that this justifies a feature set in which logging and diagnostics are supported only through component interfaces and are not integral to the Data Loader CPL. The CPL Organization also chooses to enforce this decomposition through a new architecture requirement that applies across all conformant product lines:

A CPL shall use low-level utility CPLs to implement features wherever possible.

This requirement provides a common way to track component performance across a WS via a single logging CPL. The requirement also provides improved prognostics by supporting a common health management reporting protocol. Both WS A and WS B may continue to use the existing data loader with integral logging and diagnostics, but if they do, they will not benefit from new features or improved capabilities offered in the improved Data Loader CPL.

Figure 35 shows the impact on the feature model for this scenario. The original has features that identify logging and diagnostics operations as options; the changes show features for logging-enabled and diagnostics-enabled data loaders. This change reflects the need for satisfying these features using capabilities that are not integral to the component. Rather, the component provides a capability to access logging or diagnostics data by an outside component to obtain information for logging and for diagnostics. The components may be from a CPL supported by a CPL Organization or supplied independently. The data loading component does not care as long as the component conforms to the directed use of the interface that is now an architecture requirement. WS integrators will determine how to best integrate new components that obtain the needed information to provide logging and diagnostics for use of the data loader.

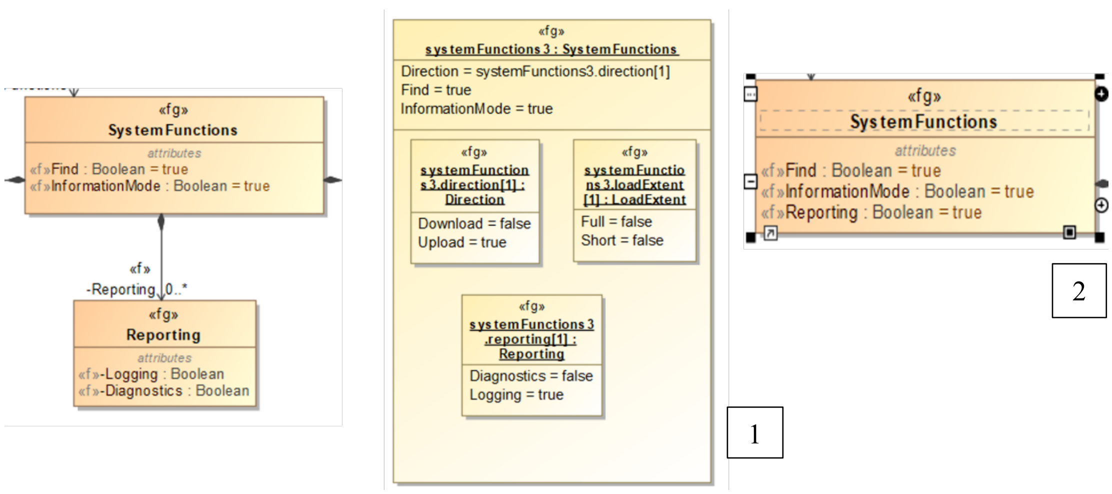


Figure 35: Change in Feature Model to Reflect CPL Change (1: Original, 2: Changed)

Model-based approaches have been prescribed as the approach to address these issues as they arise from product specification to implementation. This report examines these issues for the CPL through models that can improve communication about decisions and avoid misunderstanding, ambiguities, and incomplete product specification. The modeling approach adds feature modeling for addressing variations across instances of a product line and develops a collection of views to represent modeling representations used by component suppliers and component users, acquirers, and WS integrators.

The report describes modeling and analysis activities to support this scenario and other questions and the resulting decisions that emerge from using the 150% model. The modeling activities presented for feature modeling, feature set development, and VP assignment contribute to addressing the scenario. A variety of change management techniques should permeate the modeling approach for the CPLSM team. The collection of models in the model chain in this report reflects the connection and information flow that links the models (one-to-one or one-to-many) through the development.

4.5 Scenario 3: Application of Component Product Line to a Weapon System

This scenario has the following preconditions:

- A new WS Program Office has been created.
- The WS Program Office is early in its lifecycle, but it is already interfacing with the CPL Organization to identify potential opportunities for both organizations.
- The WS Program Office becomes aware of the available product line offerings.
- The CPL Organization gets early insight into potential changes needed for existing CPL components and learns about opportunities for new components.

Actions

- The CPL Organization reviews the new WS requirements to identify potential applicable CPLs and their existing components and feature sets.
- Some product lines will be applicable as is; their components already meet the needs of the platform.
- Other product lines may need to be modified. For example, a legacy CPL was designed to operate on a data bus compliant with MIL-STD-1533. The new platform is evaluating deterministic Ethernet data buses and considering transition to this new data transport mechanism.
- Unmet WS requirements needs may be presented as opportunities to the CPL Organization.

4.5.1 Example 1

There is a new requirement for the platform to use an ARINC Data Loader standard that can interface indirectly with the proposed Ethernet bus. Other WS components already use an alternative network service, and the WS wants the data loader to follow that deployment alternative. There may be a preexisting CPLSM for such a device, and Figure 36: Feature Selection Meeting Weapon System Needs shows the CPLSM feature set for the device. The CPL Organization informs the WS that no existing supplier provides that component. The WS acquirer will review the CPLSM and determine its suitability for fulfilling the new bus deployment requirements. The WS may seek a supplier to expand its CPL offerings to include the deployment alternative.

If no CPLSM exists, the CPL Organization could prepare a new CPLSM.

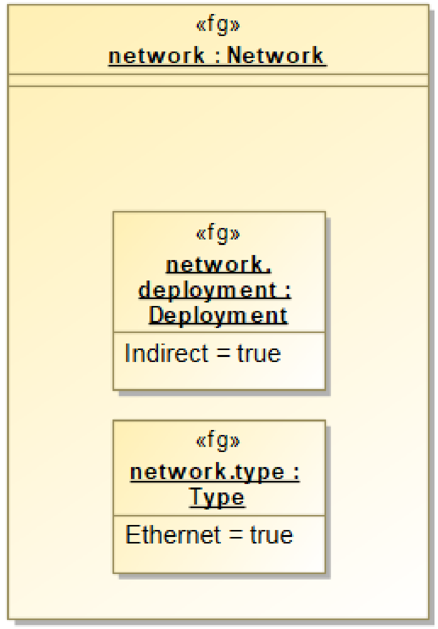


Figure 36: Feature Selection Meeting Weapon System Needs

Figure 37 shows the actions of roles in this scenario:

1. The WS acquirer studies the variation selections to determine if any of the feature sets satisfy its needs.
2. If a feature set satisfies the WS's needs, it examines built components to determine if the needed integration-ready component already exists.
3. The product line organization (champion and manager) looks at the marketplace to determine if suppliers have a component in progress to satisfy the feature set or have a closely related component that can be extended to address the feature set.

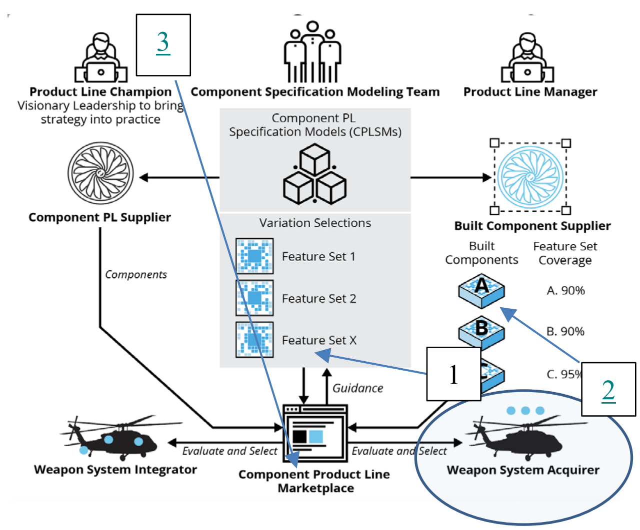


Figure 37: Illustration of Workflow for Scenario 3

4.5.2 Example 2

In this example, the WS also requires the data logging and diagnostics supported by external components. (See Scenario 2: Modify/Update an Existing Component Product Line.) The review shows that both data logging and health management are supported by components in separate CPLs. (The CPLSM has optional features to provide data used for logging and health monitoring as external interfaces. Other features provide the data loading content required for network connectivity to devices.) The WS has a need for the logging capability and health maintenance for other WS components.

The workflow for this version of the scenario (Figure 38) considers the CPL Organization the WS acquirer and the WS integrator:

1. The CPL Organization works with the marketplace to identify components that address the need for a data loader with health management and logging options.
2. The WS acquirer determines the applicability of the components that the CPL Organization identifies.
3. The WS integrator determines how to integrate the individual components to support the data loader with the desired options and how to use the logging and health management with other WS components.

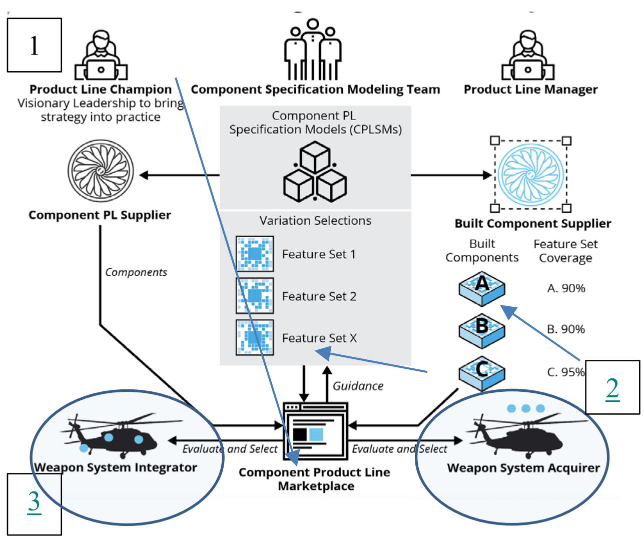


Figure 38: Identifying a Combination of Product Lines

Under this version of the scenario, the CPLSM becomes the technical specification that starts the acquisition of this new variant within the Data Loader CPL. The WS benefits in three ways:

1. It uses a component from a CPL for data loading.
2. It shares a data logging component or components across the WS.
3. It shares health maintenance capabilities across the WS via the Health Management CPL components.

5 Summary

The first report in this series established the strategy for roles and responsibilities that create and manage a CPL. This report defines a modeling approach that can be applied to specify those components. The third report focuses on governance—how decisions are made to select a product line and to determine the feature sets and how the supplier–user relationship is maintained.

This report presents an overall modeling framework and applies principles of that framework in an example. The framework looks at who creates models, how they are represented, who uses them, and how they are evaluated for use. The report introduces the concept of a model chain to describe the models used in defining a CPL. The model chain is more than a metamodel of what to represent or create. The model chain concept extends to show relationships between models, how they evolve, and the realization of architecture requirements through this creation–evolution process. This aspect of the CPL modeling approach supports the MOSA objectives of improved time to field and greater reuse of components across systems. The strategic application of reuse technology within CPL will address these objectives.

The example ARINC-615A Data Loader model is very partial in its representation, just sufficient to illustrate the model chaining from specification submodels to architecture and design views. The example includes these views to support the model understanding that a CPL implementer must have as well as creation of analysis attributes so that the supplier and potential users can assess whether and how well a CPL satisfies its specification. The analysis results are key factors in the decision-making process of component selection for integration into a WS.

This series of reports on CPL strategy, modeling, and governance provides a starting point for organizations adopting a product line approach. Next steps for developing and applying CPLs for WS acquisition include organizational transition to the CPL strategy. As described in the Strategy Report, the acquisition organizations should conduct adoption activities with stakeholders. These activities will lead to the creation of CPL Organizations to apply the approaches described in the Strategy Report and the accompanying Modeling and Governance Reports. The reports will guide CPL managers and specification teams in providing the CPLSMs to their suppliers and to WS acquirers and integrators.

Appendix A Use Case Scenario Description

Use Case Name: Perform Full Software Load

Primary Actor: Operator

Secondary Actors: Data Load Application, Target Hardware

Stakeholders: Maintenance Crew

Preconditions: Network hardware components with the expected part number must be installed on the aircraft.

Post-conditions: The Operator receives a message stating [Downloading_File_Receipt], which informs the Operator that the transfer is complete.

Trigger: An Operator invokes the Data Load Application to perform a data transfer.

Nominal Scenario:

N1. Operator (person performing the software load): The Operator communicates with the Target Hardware via the Data Loader using the Ethernet. The Operator runs the Data Load Application (DLA), which is running on the SW Loader hardware.

N2. FIND Operation: DLA Client and Server FIND Operation identifies Target Hardware on the network. The utility shows the Internet Protocol address of the hardware (LRUs) connected to the network when that particular LRU is selected on the utility.

N3. Information Operation: This operation helps in querying the current configuration details such as software part numbers and revision levels of the Target Hardware. This operation is used during on-ground maintenance operations.

N4. Initialization Step (first step): The DLA application initializes the Target Hardware Application (THA). The response to the request will be indicated by the acceptance or the refusal of this request to the DLA. If the Target Hardware refuses the request, then the DLA notifies the Operator and aborts the Information Operation.

N5. Transfer Step: If the Initialization step is accepted, the target will send the Target Hardware information to the Data Loader Protocol. In addition, the Target Hardware periodically sends a status file to indicate the status of the process.

N6. UPLOAD Operation: This operation is used to upload new or updated software, data, and configuration files to the THA. Loadable files are contained in the ARINC 665-3 Media Set.

N7. Initialization Step (first step): The DLA uses the Upload Operation. In this step, the DLA informs the THA of a request for the Upload Operation and determines if the Target Hardware is operational. The response to the request will be indicated by the acceptance or refusal of the request to the DLA. If the Target Hardware refuses the request, then the DLA notifies the Operator and terminates the Upload Operation.

N8. List Transfer Step: If the Initialization step is accepted, the DLA initiates the load list transfer by sending the Load List message to the Data Loader Protocol. The Data Loader Protocol sends the list of software/data loads to potentially be uploaded to the LRU.

N9. Transfer Step: The LRU obtains the loadable files by performing Trivial File Transfer Protocol reads of the desired files of the SW Load. The Target Hardware requests the SW Load Files. The SW files are then transferred from the DLA to the Target Hardware.

N10. DOWNLOAD Operation: This operation allows the Operator to download SW/data files from the Target Hardware using a preprogrammed list of files to be retrieved. The Download Operation works in the Media Defined Mode (MDM) or the Operator Defined Mode (ODM). In the MDM, the THA is presented with a list of files that the Operator wants to download, and the LRU then sends the files. In this mode, the DLA determines which files, from the locally stored names, are available for download. In the ODM, the Target Hardware sends the list of potentially downloadable files, and the Operator selects from the list the files that are to be downloaded.

N11. Initialization Step (first step): The DLA initializes the Download Operation. This makes it possible to inform the LRU of this operation and to know if it is operational. Optionally, the Operator can select the list of files. The DLA examines each part of the header file and provides the part numbers of the header file with download bits set to the Operator. If more than one header file with the download bit set exists on the media, the Operator may select one of these files. The access to this mode is achieved in the initialization message. The response to the request will be indicated by the acceptance or refusal of this request to the DLA. If the LRU refuses the request, then the DLA notifies the Operator and aborts the Download Operation.

N12. List Transfer Step: In this step, the Data Loader Protocol sends the list of downloadable files.

N13. Transfer Step: In this step, the Target Hardware analyzes the list of downloadable files and responds with the files defined in the list of files (.LNR file). The message will be [Downloading_File_Receipt], which informs the Operator that the transfer is complete.

Appendix B Acronyms and Abbreviations

AADL	Architecture Analysis and Design Language
ARINC	Aeronautical Radio Incorporated
AV	All Viewpoint
COP	common operating picture
COTS	commercial off-the-shelf
CPL	component product line
CPLSM	Component Product Line Specification Model
DLA	data load application
DoDAF	Department of Defense Architecture Framework
DSDM	Domain-Specific Data Model
FA	functional architecture
FACE	Future Airborne Capability Environment
FIND	Find Identification of Network Devices
FODA	feature-oriented domain analysis
GFI	government-furnished information
GOTS	government off-the-shelf
ISO	International Standards Organization
LRU	line-replaceable unit
MBPLE	Model-Based Product Line Engineering
MBSE	model-based system engineering
MDM	Media Defined Mode
MOSA	Modular Open Systems Approach
NDI	non-developmental item
ODM	Operator Defined Mode
OV	Operational Viewpoint
RFI	request for information
SV	Systems Viewpoint
SvcV	Services Viewpoint
SW	software
THA	target hardware application
UDDL	Universal Domain Description Language
VP	variation point
WS	weapon system

Appendix C Glossary

component product line
A set of components that satisfy desired capabilities and constraints (i.e., feature sets) for a range of systems

component product line manager
Operations lead for one or more component product lines to assure successful development of the product lines and use of their constituent components

component product line marketplace
The source of all component artifacts and related information reused by WS acquirers or integrators. CPL specification modeling teams and component suppliers provide their artifacts to the marketplace. Acquirers and integrators evaluate and select components from the marketplace for their WSs.

component product line model chain
The succession of models and modeling concepts necessary to support the technical basis of the component product line strategy. This model chain covers component product line development from standards to specifications to architecture, design, implementation, integration, and test.

Component Product Line Specification Model
Model that captures and represents the scope and capabilities for component product lines. Includes function, behavior, features, variations, and other elements. The models also support component understanding, selection, and tailored use by WS acquirers and integrators.

component production capability
A supplier approach that derives new products based on feature sets or feature selection. The capability includes the core assets that include development tools, methods, test frameworks, processes, and an environment to instantiate new integration-ready components from the core asset base on demand or to rapidly tailor existing components.

CPL specification modeling team
Captures and represents the scope and capabilities for component product lines in a Component Product Line Specification Model (CPLSM)

component supplier
Uses the CPLSM to identify or build components that conform to the specification. These components must be implementation ready.

feature
A user-visible aspect or characteristic of a system. Features may be required, optional, or alternative across components in a product line.

feature group
A collection of related features

feature model
A collection of features used in product line engineering to specify and communicate common and differing aspects of the products in a product line. The model organizes features to guide structure, reuse, and variation across all phases of the CPL lifecycle.

feature set
A combination of features that are required for a component to address requirements for a specific WS acquisition or development

product line champion
At the enterprise level, communicates the vision and strategy and oversees definition and scoping of enterprise product lines

weapon system acquirer
Specifies and acquires the WS that will be built through integration of components from a component product line to the extent possible

weapon system integrator
Delivers a WS to the acquirer through integration of components from a component product line with WS-unique elements

variation
The way in which two or more product variants differ from each other; the optional, alternative, or other differing features of a product that lead to individual products in a product line

References

URLs are valid as of the publication date of this document.

[ARINC 2007]
Aeronautical Radio, Inc. *Software Data Loader Using Ethernet Interface*. ARINC-615A. SAE International. 2007. https://aviation-ia.sae-itc.com/standards/arinc615a-3-615a-3-software-data-loader-using-ethernet-interface

[ARINC 2019]
Aeronautical Radio, Inc. *Loadable Software Standards*. ARINC-665. SAE International. 2019. https://aviation-ia.sae-itc.com/standards/arinc665-5-665-5-loadable-software-standards

[BigLever 2022]
BigLever Software, Inc. Concepts. *Product Line Engineering*. May 2022. https://productlineengineering.com/resources/concepts/

[Cohen 2022]
Cohen, Sholom G.; Hudak, John J.; McGregor, J.; Moreno, Gabriel; & Schenker, Alfred. *A Strategy for Component Product Lines: Report 1: Scoping, Objectives, and Rationale*. CMU/SEI-2022-SR-002. Software Engineering Institute, Carnegie Mellon University. December 2022. https://insights.sei.cmu.edu/library/a-strategy-for-component-product-lines-report-1-scoping-objectives-and-rationale/

[DMBOK2 2017]
Data Management Association. *Data Management Body of Knowledge*, 2nd ed. DAMA International. 2017.

[DoDAF 2010]
Department of Defense. *Department of Defense Architecture Framework Version 2.02*. DoD CIO. 2010. https://dodcio.defense.gov/library/dod-architecture-framework/

[FACE 2020]
Open Group. Future Airborne Capability Environment™ (FACE™) Technical Standard 3.1. July 2020. https://publications.opengroup.org/c207

[FeatureIDE 2023]
FeatureIDE. March 2023. https://featureide.github.io/

[Felfernig 2022]
Felfernig, Alexander; Ortner, Bettina; & Le, Viet-Man. 2022. Table-Based Knowledge Representations for Industrial Feature Models. Pages 245–248. In *Proceedings of the 26th ACM International Systems and Software Product Line Conference – Volume B*. ACM. September 2022. https://doi.org/10.1145/3503229.3547047

[ISO/IEC 2015]
International Organization for Standardization and International Electrotechnical Commission. *Software and Systems Engineering—Reference Model for Product Line Engineering and Management*. ISO/IEC 26550:2015. ISO. 2015. https://www.iso.org/standard/69529.html

[Kang 1990]
Kang, K.; Cohen, S; Hess, J.; Novak, W.; & Peterson, A. *Feature-Oriented Domain Analysis (FODA) Feasibility Study*. CMU/SEI-90-TR-021. Software Engineering Institute, Carnegie Mellon University. November 1990. https://insights.sei.cmu.edu/library/feature-oriented-domain-analysis-foda-feasibility-study/

[Kruchten 1995]
Kruchten, Philippe. Architectural Blueprints: The "4+1" View Model of Software Architecture. *IEEE Software*. Volume 12. Issue 6. November 1995. Pages 42–50.

[MBPLE 2021]
No Magic, Inc. *Model-Based Product Line Engineering*. 2021. https://docs.nomagic.com/display/PLE2021xR1/Model-Based+Product+Line+Engineering

[Open Group 2023]
The Open Group. *Open Universal Domain Description Language (Open UDDL), Edition 1.1*. 2023. https://publications.opengroup.org/c198

[Padilla 2019]
Padilla, M. O.; Davis, J. B.; & Jacobs, W. Comprehensive Architecture Strategy (CAS). The Open Group. September 2019. https://www.opengroup.us/face/documents.php?action=show&dcat=87&gdid=21082

[Schenker 2023]
Schenker, Alfred & Cohen, Sholom G. *Component Product Line Governance*. CMU/SEI-2023-SR-009. Software Engineering Institute, Carnegie Mellon University. May 2023. https://insights.sei.cmu.edu/library/a-strategy-for-component-product-lines-report-3-component-product-line-governance/

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
(Leave Blank)	May 2023	Final

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
A Strategy for Component Product Lines: Report 2: Specification Modeling for Components in a Component Product Line	FA8702-15-D-0002

6. AUTHOR(S)

Sholom Cohen, John Hudak, John McGregor

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213	CMU/SEI-2023-SR-008

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
SEI Administrative Agent AFLCMC/AZS 5 Eglin Street Hanscom AFB, MA 01731-2100	n/a

11. SUPPLEMENTARY NOTES

12A DISTRIBUTION/AVAILABILITY STATEMENT	12B DISTRIBUTION CODE
Unclassified/Unlimited, DTIC, NTIS	

13. ABSTRACT (MAXIMUM 200 WORDS)

This report is the second in a series to establish a strategy for creating and managing a component product line (CPL), to define a modeling approach for specifying components, and to describe a governance method for making decisions to select a product line, determine the feature sets, and maintain the supplier–user relationship. The strategy considers who creates models, how they are represented, who uses them, and how they are evaluated for use. This report introduces the concept of a “model chain” to describe the models used in specifying a CPL. The model chain concept extends to show relationships between models, how they evolve, and the realization of architecture requirements through the creation–evolution process. The report applies the principles of this framework to three example governance scenarios based on a specification of a common industry standard. The examples include specification views to support the model understanding that a CPL specifier must provide and to create feature, function, and other model analysis attributes so that the supplier and potential users can assess whether or how well a CPL satisfies its specification. The analysis results become key factors in the decision-making process of component selection for integration into a weapon system.

14. SUBJECT TERMS	15. NUMBER OF PAGES
governance, model chain, product line, software modeling, weapon systems	83

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. Z39-18 298-102