

3.4.3 Product Family Engineering

The product family engineering model specifies a product family in the form of a generalized product model. An adjunct to this model is a mechanism for deriving instance product models from the representation of the product family. The process engineering model provides support to product manufacturing projects for specifying and deriving an instance product model using the product family engineering model.

A product family is conceived based on a perception that its instances are substantively similar, differing as a consequence of how deferred decisions, as specified in the decision model, correspond to different instance products. To support this, a product model corresponding to a single canonical product is generalized with annotations that specify the implications of differently resolving those deferred decisions. Full resolution of all deferred decisions determines a subfamily consisting of a single product; any partial resolution of deferred decisions corresponds to a product family model that represents a subfamily consisting of multiple products.

The product family engineering model differs from the software product model in three respects: (1) it elevates focus from the project-customer level to the program-market level, (2) it provides an aggregate representation of a product family as a whole, along with an associated means of deriving an instance product model from that representation, and (3) it encompasses not only software engineering but also systems and hardware engineering considerations as needed to realize a complete product.

A feasible product is one for which all specified deferred decisions have been resolved, resulting in a singleton subset of the product family. Multiple feasible products are determined if some decision resolutions are incomplete, representing uncertainty as to which resolutions will determine the best product for specified customer needs.

Partially resolved decisions determine a subset of the product family, corresponding to multiple potentially feasible products. It is possible to derive multiple products for a comparative evaluation of these as alternatives. It may be possible to defer some decisions for resolution during operational use of the product. It may also be that the

domain is incomplete for addressing specific customer needs, requiring either relaxation of those needs or enhancement of the domain.

In adopting the product model form defined for basic software product engineering, the product family engineering model would have the same seven facets and associated elements: requirements, environment, design, analytics, components, verification, and delivery. (Depending on the domain and appropriate or preferred practices, model facets and elements could be added, modified, or removed.)

Representing a Product Family

A product family is an aggregate representation of an envisioned set of products having similar capabilities but differing in aspects of their observable behavior. The basis for a product family model is a notional instance product model, made to represent the family by annotating it to show the differences among the product models for all derivable instances. Differences among instance products are characterized by annotations that show the effects on the model of differently resolving deferred decisions.

Decisions span a product family model—each decision can affect the content of multiple elements within any or all facets of the product family model. Consequently, each element of a derived product model may be determined by the individual resolutions of disparate decisions; similarly, the content of disparate elements of a model can be influenced by how a given decision has been resolved.

The motivation for creating a concrete realization of a product family is to provide a medium from which customized instance products can be derived by resolving deferred decisions without repeated effort to construct any common portions. Decisions are referenced in the form of annotations within the product family model that indicate how different resolutions cause product model content to differ. These annotations can be resolved manually or using an adaptability mechanism (for example, as described in section 4.1).

Any partial resolution of deferred decisions associated with a product family corresponds to the definition of a product subfamily. The instances of a subfamily are

distinguished by the remaining—partially or fully unresolved—decisions. A product family model for which all associated decisions have been resolved is the product model for a specific instance of that product family.

Incrementally Extending a Product Family

In practice, a product family will describe a set of products of which only a subset is practical to build. Domain management, based on market circumstances, will prioritize supporting product capabilities that are likely to be most eminently needed, with plans to expand such capabilities as program resources permit.

Typically, initial increments will focus on assimilating capabilities of previously developed products to the degree these align or can be modified to align with the domain definition, and extending these with decision-based annotations to accommodate identified opportunities for relevant variability. Having defined a product family model that is a sufficient basis for building products that will reasonably support current market needs, subsequent increments will focus on adding omitted capabilities, improving supported capabilities as market needs evolve, and expanding variability to increase buildable domain scope.

A family may in principle include instances that are not viable to construct in practice (i.e., being equivalent to a subfamily). When the need for an excluded product is found, this may motivate revising the domain to enable building such a product. Alternatively, or in the interim, it may suffice to build another instance of the family that is a sufficiently close approximation of the envisioned instance.

Verifying a Product Family

Verifying a product family entails evaluating both consistency with the domain definition, particularly the decision model, and consistency among all elements of derived product models.

With respect to the domain definition, the product family model will typically be only partial, supporting the derivation over a restricted domain scope, but iteratively extended as market needs and domain resources warrant. The product family model must be consistent at the least with the limited domain definition that corresponds to

the domain's currently restricted scope but may exceed this scope in preparation for an envisioned extended domain scope.

Verification can occur in some combination of three approaches:

- Direct inspection – As a routine aspect of building the elements of a product family model, each element is reviewed both as being conformant to expected product model content and with respect to its accommodation for applicable decisions that express differences among the derivable instances of that element
- Selective instantiation – Instances of a resolved decision model are conceived and applied to the product family model to derive instance product models; each of these are individually evaluated both as proper to the associated resolution of the decision model and as being a consistent and complete product model
- Family-level analytics – *{extension of product analytics model; addressed separately in Chapter 5 as predictive analytics and otherwise}*

The process engineering model should provide mechanisms that elevate the capabilities for product family model evaluations within a more expansive process of product manufacturing.