

3.4.2 Domain Definition

The domain definition model specifies the nature and extent of the product family represented by the domain. This model has five elements: a synopsis that characterizes the essential purpose that every encompassed product will satisfy, a catalog of relevant terminology, a catalog of legacy knowledge, assumptions that characterize similarity among encompassed products, and a set of deferred decisions whose resolution are sufficient to reduce the product family to a specific product.

The domain definition serves three purposes:

- As an informal characterization by which developers maintain a shared understanding of what sort of products are needed by a market;
- As a basis for determining whether capabilities needed by a customer are likely to correspond to an instance of the product family;
- As the conceptual foundation for defining a product family and an associated means for deriving customized products.

Synopsis

The domain synopsis element is a concise description of all products within the scope of the domain. This description uses domain terminology to express the abstraction that the product family represents. The synopsis is sufficient as a description of any individual product that is within the scope of the domain. Any product that fails to adhere to the synopsis in every aspect is excluded from the product family; the synopsis can be modified if the membership of the family is found to be too broad or too limited. The synopsis is elaborated as assumptions of commonality that are applicable to all instances of the product family.

Terminology

The domain terminology element specifies a common vocabulary for describing the problems and solutions that a product family addresses. This element establishes a

uniform market-encompassing terminology for the domain, which supports an understanding by developers of how customers perceive their endeavors.

It further identifies and correlates any differing nomenclatures that different customers are found to use in discussing their endeavors such that there is a definitive translation between the common vocabulary and any differing customer nomenclature.

Legacy

The domain legacy element identifies relevant sources of information that serve as the foundation on which the domain is built. The initial motivating basis for creating a domain, beyond the existence of a perceived market for its products, is accessible developer competence for the development of the envisioned product family. Legacy content includes any relevant scientific, technical, or market references, any previously developed products or components that fall within the scope of the product family, and any material that may be useful in understanding future capabilities that products may need to support.

Assumptions (of Commonality and Variability)

The domain assumptions element elaborates the domain synopsis to provide a market-specific description of how products that belong to the envisioned product family are similar. Assumptions of commonality informally characterize how all instances of a family are alike and therefore differ from excluded instances. Assumptions of variability informally indicate the various ways that any two instances may differ from each other.

Commonality assumptions provide an intuitive sense of the sort of products that are properly within the buildable scope of the domain. Based on these assumptions, any envisioned member of the set is an equally good representative for the general purpose of all products in the set. The basis for deciding which member of the set is best for a specific purpose is expressed in the associated variability assumptions. These assumptions are formalized in the domain decision model element as the basis for the systematic derivation of specific products.

A primary set of commonality assumptions characterizes the set of products that the domain synopsis identifies as being a product family. An associated set of variability assumptions defines the ways in which these products can differ.

Variability assumptions describe criteria by which a family can be partitioned into subfamilies, with each subfamily having a membership that is more similar and therefore more limited than the containing family. Instances that satisfy the same resolution of a variability assumption will share additional assumptions of commonality, requiring additional assumptions of variability to distinguish among them. (Instances of a family that cannot be purposely distinguished by any variability assumption are considered equivalent and therefore effectively interchangeable with respect to the addressable problem-solution.) Repeatedly applying this partitioning to subfamilies will produce a hierarchy of subfamilies that provides useful insight into the nature and composition of market needs motivating the domain.

As an elaboration of the domain synopsis, domain assumptions provide a still informal but more detailed basis for determining the feasibility of deriving a product that will fit a given customer's specific needs. A potentially feasible product is one that satisfies the commonality assumptions of the domain and can be distinguished from other similar products entirely by reference to variability assumptions associated with the domain. For a given set of assumptions, a brief analysis will suffice to determine whether initiating a product manufacturing effort is warranted. In some cases, the analysis may conclude that such an effort is not justified or may expose a need to refine the domain, as envisioned or as built, by revising these assumptions appropriately.

Decision Model

The decision model element is a formalization of the alternatives, as expressed in variability assumptions, that a developer must resolve to characterize and build an instance of the product family. These alternatives are coalesced into the form of deferred decisions that in being fully resolved correspond to a single buildable instance of the product family.

Each deferred decision is expressed as a choice related to aspects of product behavior that are meaningful to a customer such that the basis for its resolution in their needs can be understood by the customer. A developer may need to work with the customer in exploring alternatives and tradeoffs for each decision resolution to establish an understanding of the implications of such resolution on product behavior.

a concern that is relevant to and can be understood and answered by a customer regarding their needs

Certain decisions will be independent, being resolvable in any order. Decisions that are logically related, often being traceable to a single or related set of commonality assumptions, may be organized into a decision set that can be resolved as a unit. Dependencies may exist among some decisions (e.g., the particular resolution of a dominant decision may constrain which subordinate decisions need to be resolved); in such cases, the decision model will define a partial ordering for resolution of these decisions.

Under DsE, the decision model that characterizes a product family is a primary organizing medium for domain engineering and the essential framework for product manufacturing. Alternative resolutions of each deferred decision partition the product family into mutually exclusive subfamilies. A progressive resolution of deferred decisions reduces the candidate set of derivable products until a non-reducible subfamily is determined, corresponding to a single derivable instance of the characterized product family.

Characterizing a Decision

(some of this belongs in 3.4.4)

A decision is defined according to what sort of content is appropriate for its resolution. The resolution of a decision can take any designated form, constrained as appropriate in its representation and its usage in resolving a particular variability among products (e.g., computing a runtime data value, filling in customer-specific product nomenclature or explanatory content, defining a specialized processing component).

Implicitly, every decision being an unresolved uncertainty is initially undefined. The means of resolving decisions is specified in the processing engineering model. Its resolution can be restricted in the form of its content but can be reverted at any time to undefined:

- Fixed Value (limited to a specific resolution pending future support for alternatives)
- Default Value (typically preferred among alternative resolutions)
- Developer-provided (a direct expression of a typed value that specifies some aspect of customer needs)
- Direct Value (a value of the specified data type, such as boolean, numeric, enumerated, text string, graphic or animated image, audio or video)
- Direct Computation (a source or object implementation of a specified fixed or adaptable component interface)
- Derived Selection (a computation based on other (potentially changing) decisions)
- Multi-valued (a decision of any form can be specified as permitting multiple values, and those being ordered by some criteria or unordered, corresponding to iteration within relevant elements of the product model)
- Conditional (combining or selecting among other forms of resolution)

Correspondence to Domain Assumptions

Decision Constraints

A decision will generally be constrained as to the values it can be given, including any dependencies on the resolutions of logically related decisions. The resolutions of two related decisions may be mutually dependent, even to the point that the resolution of one determines the resolution of the other.

Composite Decisions

A composite decision is one that is comprised of multiple related decisions that must be consistently resolved as a whole. The resolution of each of these decisions may be constrained in the same way as any other set of related decisions.