3.4.4 Process Engineering

The process engineering model specifies the process to be followed by a program's product manufacturing projects to build a product model for a deployable product. A product manufacturing process is a formulation for the managed specification, generation, evaluation, deployment, and evolution of instances of an associated product family. An adjunct to this specification is provision of a corresponding product manufacturing capability that supports the performance of that process.

A manufacturing capability is an assisted environment in which product developers are able to perform the specified manufacturing process to build customized instances of the product family. This capability is not intended to be an efficient or effective means for building products outside the scope of the product family. The narrower the scope of this capability, the more efficient it will be within that scope and the more it will inhibit building anything outside that scope.

The essence of product manufacturing is the development of a product specification. A product specification is the medium by which developers resolve the deferred decisions specified in the domain decision model as needed to characterize a product that suits a given customer's specific needs. A product model defining a particular instance product is then derived by applying a product specification to the product family:

[product family (product specification) == product model] –> product

Any partial product specification determines a "candidate subset" of the supported product family—those products that are a potential fit to the customer's needs. The candidate set changes as deferred decisions are resolved or modified. A complete product specification designates a singular subfamily, identifying one product that is the best buildable fit to a given customer's needs.

Product Manufacturing Process

The product manufacturing process element specifies the capabilities needed by projects to perform product manufacturing for the domain, referencing the decision model and product family engineering model elements. A product manufacturing

process has an analogous objective to basic software-based product development—timely delivery of a product that meets the needs of a customer—but with improved quality and reduced effort to build. DsE product manufacturing should in some form support six essential capabilities[1]:

- Project management – the orchestration of the incremental and iterative performance of a process for realizing a product customized to a customer's specific needs
- Product specification – the resolution of deferred decisions to characterize a problem-solution that corresponds to the essential needs and circumstances of a customer
- Product evaluation (validation) – the evaluation of the degree to which a product specification is consistent and complete relative to the current understanding of customer needs, given domain capabilities
- Product realization – the derivation from the product family model of a product model corresponding to a given product specification
- Product evaluation (verification) – the evaluation of the degree to which a given product model is a proper realization of a given product specification
- Product delivery – the deployment and support of a product for operational use by a customer

The project management and product delivery capabilities of product manufacturing are essentially equivalent to the corresponding facets of a software product engineering project (e.g., as described in chapter 2). The product realization capability is the means by which a manufacturing project applies the product family engineering mechanism to derive a customized instance product model (as discussed in section 3.4.3). (The nature of product specification and evaluation capabilities are discussed in below subsections.)

[1] The five conceptual elements of product manufacturing, as described in section 3.3, are a direct formulation of these capabilities.

Specification of a particular manufacturing process is guided by the level of process capability targeted in the domain strategy element of the program management model, limited by the level of resources allocated by the program for developing a complementary manufacturing capability. The above essential capabilities of product manufacturing align to the highest level of process capability, targeting a streamlined process that enhances the productivity and product quality of a program's product manufacturing projects. To the degree that the domain strategy targets a lower level of capability, the specified manufacturing process will more closely resemble a conventional development process, requiring less investment in process engineering but requiring a correspondingly increased effort by manufacturing projects.

Necessarily, the manufacturing process may be refined progressively over time as the domain evolves to cover its envisioned and changing domain scope and adequate support for all elements of a product model. However, it may also be necessary, particularly early in the development of a domain, to include capabilities in the process for conventionally developing parts of a product model that are not yet adequately supported—covering either incomplete, overly diverse, or newly emerging aspects of the domain. This might, for example, support completion of components that the product design accommodates with defined interfaces but that are not yet implemented in the product family or only partially for some deferred decisions.

Modifying a generated product may serve as an expedient for building an otherwise excluded product but can be difficult not just initially but over time as a customer's associated needs change. A better alternative is to initially build only a close approximation to the needed product and then extend the domain to encompass that product's missing capabilities. In fact, a domain is meant to evolve over time, not only to extend the product family to permit building previously excluded products but also to accommodate changes in the market composition and its aggregate needs.

{address hdw/sw codesign perspective, as factored into process engineering (defining the manuf process, including any hardware fabrication and assembly, for both hdw & sw components)}

Expressing a Product Specification

A product family defines a candidate set of buildable products for customers in a domain-targeted market. A developer creates a product specification by resolving deferred decisions to describe the candidate that best fits a given customer's particular needs and circumstances. The decision model defines these decisions and the form each will take in a product specification, providing a shared medium in which product family engineering and process engineering express how a needed product is distinguished from other instances of the product family.

The product specification capability includes assessing the status of consistency among decisions, verifying that each decision has a valid resolution for its specified form and noting the degree to which decision-associated consistency criteria are satisfied (e.g., flagging any discrepancies for resolution). The product realization capability may be applied to an incomplete product specification (to generate a product subset) but only if it satisfies all relevant consistency criteria

The process engineering model determines the form in which each decision, consistent with its definition in the decision model, is presented to a developer for completing a product specification. A partially resolved product specification characterizes a subset of the product family, corresponding to those products that are a better fit than others to a customer's needs. Through continued resolution of deferred decisions, the candidate set can ultimately be reduced to a single product.

In principle, developers can resolve decisions in any order. However, process engineering may impose a partial ordering on decision resolution to reflect logical dependencies among decisions or to provide for a more structured and therefore simpler process.

In general, the decision model formulates decisions that are related aspects of the same subject matter as elements of a composite decision. The resolution of a composite decision (including the decision model as a whole) may be partitioned according to process-organizing criteria (eg, per work product, per customer process phase or user role, or per developer areas of competence or delegated responsibility).

Beyond defining each deferred decision, the decision model also defines, via conditional constraints associated with each decision, dependencies among decisions. Some decisions can be resolved only if one or more other decisions have either already been resolved with acceptable values or have not been resolved. Similarly, resolving a given decision may for consistency constrain the resolutions of other decisions including causing their existing resolutions to be changed.

The process engineering model determines how decisions can be resolved, consistent with the decision model:

- A decision can initially have an unknown value, a preferred default value, or a value determined based on the values of related decisions.
- A decision's value can be changed based on some combination of developer action and computations based on the values of specified related decisions; a decision can be changed to unknown or its default value by these same means.
- A decision's value may be constrained if product family engineering has not provided support in the product family model for some decision model specified options.
- Developers may be limited in modifying a decision's value—setting, changing, or reverting to unknown—due to associated decision model specified constraints.
- A decision can be designated as accommodating multiple candidate values in its prescribed form (e.g., to allow for retaining alternate values to be compared or combined for a final resolution).
- A decision can have associated metadata for establishing the provenance of its value, such as the timing, sources, and degree of confidence/certainty in related decisions' values as the basis for its determination and to be given as rationale for its value (e.g., to be used in combining or choosing among alternative candidate values).

Given uncertainty in how to resolve some decisions, it may be necessary to derive a set of candidate products for different resolutions of those decisions and comparatively evaluate them for a best fit to a customer's perceived needs. Alternatively, support may be provided to translate uncertain decisions into operational options that can be decided during installation or dynamically by users of the product. After further experience with the product, it may be possible to more definitively resolve these decisions, resulting in a revised product having more efficient deterministic behavior.

Evaluating a Product Specification

The evaluation of a product specification by a product manufacturing project has two aspects:

- Validation – determining whether the specification is a proper expression of customer needs within the bounds of the domain
- Verification – determining whether the derived product model is a proper realization of the specification

Validation relies primarily on direct market-, customer-, and domain-cognizant reviews of the in-work specification as to its consistency with expressed customer needs. Verification focuses on the examination of derived product model elements (including specifications, deliverable documents, and software) to ensure consistency with the product specification. Defects found in verification may be due to errors in the product specification or in the product family model. In addition to reviews, verification may include static and dynamic analyses based on analytics facet content and testing based on verification facet content of the generated product model. Typically, interim versions of a product specification can be used to derive only partial product model materials that may support early validation and verification efforts.

The product realization capability applies a consistent and sufficiently complete product specification to the product family to derive a corresponding customized product model. A product model includes the product in deployable form with documentation, verification materials, and, possibly, a facsimile of the product's operational environment. Supported evaluation capabilities could also be included as

an ancillary element of the product delivery for use in supplementary customer acceptance, validation, and certification of the as-built product for operational use.

Evaluating Alternative Specifications

Typically, a developer with relevant market and technical competence will be able to resolve many deferred decisions based on an exploratory examination of customer needs and circumstances. Other decisions may require consideration of alternatives, with analyses and comparisons of the differences that each will have on resulting product behavior. This may further require consideration of how each alternative interacts with the resolutions of other decisions (based on dependencies among decisions that constrain their resolutions). A developer may need to explore with customer representatives how to expeditiously resolve these alternatives and associated tradeoffs.

In some cases, comparative empirical evaluations with alternative resolutions of one or more decisions may be needed to determine the best combination given the customer's needs. This may require identifying alternative product specifications, differing in their resolutions of uncertain decisions, from which corresponding product models are then derived. For these cases, the process engineering model should accommodate specification versioning and associated mechanisms that support the capability for comparative evaluations among these alternatives. This could require the tentative resolution of uncertain decisions or possibly supporting the derivation of multiple product models based on uncertain decisions that have each been resolved to a limited number of alternative candidate resolutions.

A decision may have alternative candidate resolutions resulting from developer uncertainty or due to related decisions having alternative resolutions. It may be possible to assign a level of confidence based on the metadata associated with alternatives, as criteria for deciding among them. A developer-specified resolution that creates a conflict among related decisions may be overridden or require changing those related decisions to establish consistency among them. In some cases, this may require further analyses of how alternative resolutions affect the resulting product.

Product Manufacturing Capability

The product manufacturing capability adjunct element takes the form of a domain-specific environment—a software-based product that constitutes the "domain infrastructure" for performing a specified product manufacturing process. This infrastructure provides definitive guidance on performance of the manufacturing process that, depending on the level of program investment in domain engineering, can be enabled with an appropriate level of automated support for that process. For a domain whose products include specified hardware components, this guidance and associated support may address the acquisition of commercially available hardware or of the means to fabricate and assemble specialized hardware.

The building of automated support for product manufacturing is a variant of a software development project for which a domain-specific environment is its product. A domain-specific environment enables, through the provision of appropriate mechanisms and raw materials, the streamlined building of customized products represented by a product family. This element specifies the mechanisms that support performance of the process as a whole, with the exception of product realization which the product family element specifies in terms of the means and raw materials provided by a concretely realized product family.

For product specification in particular, resolving decisions can be entirely manual but is easily supported with simple automated tools. Decisions may be resolved, for example, via a prescribed dialog, with the conditional filling of decision resolutions in various types of linear, tabular, or hierarchical forms. Alternatively, more complex forms of decision resolution such as constructive domain-specific product modeling can be envisioned and supported. In whatever form obtained, resolved decisions can then be applied for the realization of a customized product model including all materials needed for computation, documentation, evaluation, and delivery of the specified product.