Modeling Variation in Production Planning Artifacts

Gary J. Chastek
Software Engineering Institute
gjc@sei.cmu.edu

John D. McGregor
Clemson University
johnmc@cs.clemson.edu

Abstract

Production planning and variation modeling are interdependent parallel activities critical to the success of a software product line. Software product line organizations design a production capability to satisfy their business goals. That production capability is dependent on and must support the full range of product variation. Current techniques for variation modeling identify and handle variations among products but fail to recognize variations that result from business goals such as rapid time to market which are satisfied by how products are built. In this paper we present a view of our production planning technique and describe our preliminary research into its relation to variation modeling.

1. Introduction

Variation distinguishes a software product line from single product development. A variation is any concern that will vary from one product to another or that may vary in time. Product feature models alone are inadequate for modeling the full range of variations the software product line organization must manage. They capture product but not production variations. Goals like mass customization are production rather than product issues, and can yield strategically-significant variations. The variations in how products are built are a direct result of the goals of the organization. For example, a business goal to compete in the global market can lead to variations in testing processes used depending upon the market for which each product is intended.

These variations are the result of strategic decisions made during early product line planning activities of the "What to Build" (WTB) pattern [1], shown in Figure 1. Scoping uses the marketing analysis to identify the members of the product line. During this activity variations in features among the products are identified. The market analysis identifies market segments and analyzes the differences among them. Building the business case requires the creation of a justification for the use of the organization's assets to create and operate the product line organization. It also provides the opportunity to identify additional variations that are not directly related to product content but that are related to product production. We will call the variations found during the WTB activities strategic variations. A change in a strategic variation results in a change to at least one of the WTB artifacts, and vice versa. For example, expanding the scope of the product line may result in additional test requirements if the additional product is intended for a new market.

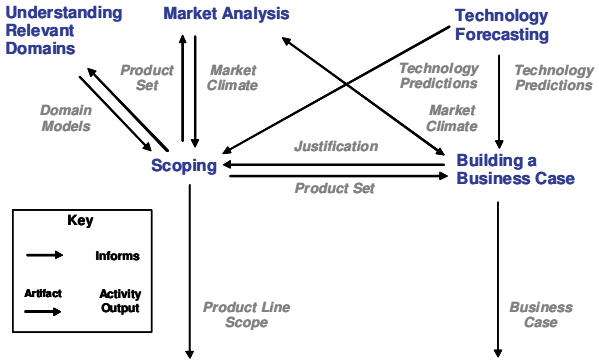

Figure 1 What to build pattern

In addition to strategic we also define tactical variations as those variations that arise from the resolution of strategic variations. For example, the development process will have a step in test planning where the levels of test coverage are determined by the intended market. Tactical variations typically do not necessitate a change in a strategic variation but could if, for example, the choice an architectural mechanism provided the possibility of supporting a wider range of products.

The progression from strategic to tactical variation and ultimately to variation point corresponds to positions along the Variation axis in the space defined in [2], as illustrated in Figure 2. The strategic variations

represent "what concerns" are important and are placed further out on the Variation axis than the tactical variations that represent "how" those concerns are addressed. Similarly, tactical variations are placed further out than variation points since tactical variations represent the broad concern while a specific variation point addresses only a portion of that concern. Finally, a bound variation point represents a fixed point in the two-dimensional single-product plane. This progression corresponds to the hierarchical separation of concerns described by [3].

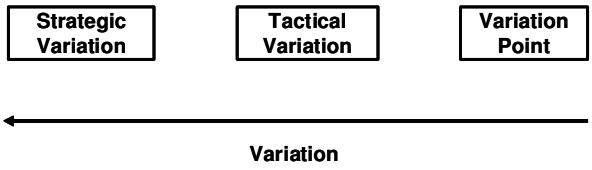

Figure 2 Variations in context

The goal of this paper is to describe our preliminary research into the relation between variation modeling and production planning. Its contribution is to identify key software product line variations not currently considered (i.e., variations related to how products are produced[1]) and to extend the work of Berg et al [2] by explicitly identifying strategic and tactical variation points.

The subsequent sections describe our work in production planning, variations in the production system and their relation to variation modeling in general, and finally the current directions of our ongoing research.

2. Production Planning Artifacts

How a software product line organization builds its products is a system, the production system [5], that has both functionality (e.g., the development tools employed) and quality attributes (e.g., how quickly a specified product can be delivered). Production planning devises a production system that

- Satisfies the organization's goals and constraints for its product line
- Coordinates the design of the core assets with the production system
- Communicates the effective use of the production system to the product developers

This is achieved by formulating a production strategy [6], constructing a production method [7], and documenting the production process in a production plan [8]. The production strategy links the business goals of the software product line organization to its means of product production. Porter's Five Forces model [9] is used to derive strategic actions that the production system must provide. For example, the organization can resolve the force of potential entrants (i.e., new competitors) into the market by taking the strategic action of automating product production and thereby reducing the per-product costs.

The production method defines an overall implementation approach that coordinates the efforts of the core asset and product developers. Method engineering techniques are used to define a method specifically for the needs of the product line organization. The production method contains processes, tools, and models that are used to implement the strategic actions related to product production. Efficiencies in production can be achieved by the elimination of inconsistencies across the processes, models, and tools. For example, the automating product production strategy can be partially implemented by using automated test generation tools.

The production plan communicates the production process details to the product developers. A generic production plan is developed as a core asset. Each core asset has "an attached process that specifies how it will be used in the development of actual products." [1] The plan is made product-specific by adding in the attached processes from the core assets selected for use by the product builders. For example, the production plan would specify the parameters to be used in instantiating the test generation facility.

The representation of variation points in the production planning artifacts varies by organization but should be compatible with the approaches being used in the organization's core assets. Ultimately the variation point representation must support the production of products in accordance with the goals for the software product line.

3. Production Planning Process and Modeling Variations

Each of the artifacts produced during production planning provides a different view on the variation model. During the early product line planning activities, described in the WTB pattern, variations are identified based on the differences among product feature sets and the differences in production techniques as identified from the business goals and

[1] Schmid and Eichelberger have addressed binding time as a meta-variability using aspects [4].

market analysis. These strategic variations are a primary input into the production planning process. As production planning moves from strategy development to method development to plan construction, the view of production becomes more concrete and focused.

3.1 Production Strategy

The production strategy defines the overall approach to producing products and begins the production planning process that ultimately results in the resolution of variation points. The strategic variations, identified during the analysis of the artifacts that result from applying the WTB pattern, lead to the identification of tactical production issues. For example, the strategic variation of addressing the delivery needs of a diverse market might translate into a tactical variation in the production process, e.g. waterfall vs. agile, that would be selected by the product builders for a specific product.

Table 1 Porter's forces

Porter Forces	Strategic Actions	Example impacts on variation
Potential Entrants	Leverage economies of scale	Minimize variant choices to get maximum use from each variant
Substitutes	Raise the cost of switching to another product	Provide some minimal variant implementations for each variation to allow for a low-cost product to attract or retain customers
Suppliers	Commoditize required components	Implement variations behind standard interfaces to maximize the number of potential suppliers
Buyers	Differentiate from other products	Maximize the number of variations to provide flexibility
Competitors	Improve features	Expand the number of variants available per variation to add features for customers

During production strategy development, strategic actions are identified as a means of resolving each of the forces identified using Porter's Five Forces strategy development model. The strategic actions that resolve one force can be at odds with another action resolving a different force leading to tradeoffs between those actions. Table 1 shows examples of strategic actions and corresponding impacts on tactical variations.

For example, the Potential Entrants force can be resolved by the strategic action of leveraging economies of scale, implying a minimization of the number of variant choices. Similarly the Buyers force can be resolved by differentiation from other products, leading to the maximization of the number of variations. This conflict can be resolved by maximizing the number of variations but minimizing the number of variants for each variation. This provides flexibility without a large upfront investment with the variation point providing the option to later expand the number of variants available.

3.2 Production Method

Method engineering designs constructs and adapts processes, models and tools for the development of software systems [10]. Method fragments are coherent pieces of development methods. The method engineer considers the characteristics of the organization and the current project to define a development method that will meet the specific needs of the project. In a software product line organization this includes the business goals and the variations in how the products will be produced.

The production method specifies how products will be built and directly affects how core assets are designed to support product building. The decisions about how to resolve the tactical variations are part of the method engineering activity that defines the production method. The production method uses a model of the tactical production variations to identify points of variation in the assets, see Figure 3.

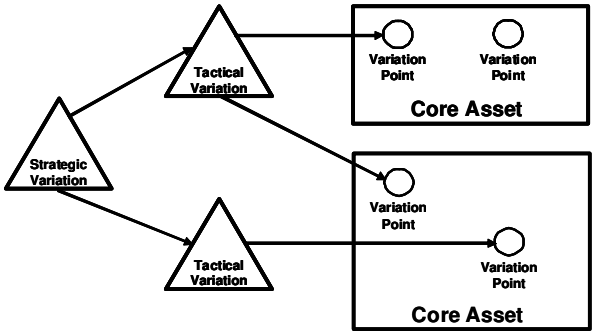

Figure 3 Variations to variation points

The model consists of the decisions that must be made regarding which portions of the production method to use in a specific situation. For example, if multiple deployment platforms are targeted, the

product builder would need to switch between different memory models and the tools specific to each. The production method would capture this variation in a process description and would specify the tools and models for both platforms.

The method engineers responsible for creating the product method determine the set of variation mechanisms from which the asset developers will be allowed to select for implementing their products. For example, aspect-oriented programming may be identified as a useful technique to include in the production method. Core asset developers will be able to create an asset that includes a basic unit and a set of aspects from which the product builder may select. The attached process for that asset would provide instructions for using the aspect weaver and other related tools and for incorporating the actions related to aspects into the production method.

The production strategy is a high-level answer to the question: "How can product development satisfy the organization's goals for the software product line?" The strategy provides a direct link between the product line goals and the means of product production. For example, a product line goal of "faster time to market" could lead to a strategy in which automation is used wherever possible.

3.3 Production Plan

The production plan guides product builders through building their specific product using the core assets. There are two "flavors" of the production plan: the generic and the product-specific.

The generic production plan is largely constructed from method fragments which correspond to the production variations. The method fragments are coordinated with the attached processes of specific core assets. The generic production plan, which is a core asset, is incomplete because it includes a model of the product variations. The variation points in this model represent decisions to be made by the product builder.

The generic production plan is instantiated into the product-specific production plan using the information in the attached processes of the core assets selected to satisfy the product variations, see Figure 4.

When a specific core asset is selected for use in building a product, its attached process is added to the emerging product-specific production plan. That includes adding a method fragment to the production method for that product just as the information provided by the core asset is added to the product. The generic production plan provides the context for making the tactical decisions required to select a variant from among those possible at each variation point. The fully instantiated product-specific production plan has no unbound variation points, but the product may still have unbound variation points.

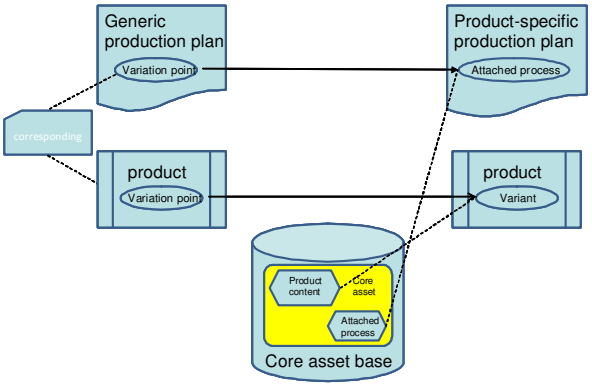

Figure 4 Production plan instantiation

3.4 Example

Bundling is a business strategy that groups a set of the products in the product line for sale as a unit. The products vary from each other but share a common high level purpose. A suite of word-processing, presentation graphics, and spreadsheet products are bundled based on a user's role in an organization. A security package bundles several products that all relate to the same general purpose - ensuring the security of a computer. The distribution of variants for a single feature across the products in the unit is a strategic variation. Some specific strategic variations include the range of operating systems to be supported or the variety of human languages to be supported.

The strategic variation of having different functionality based on a common purpose leads to a number of tactical variations. For example, file formats vary because different users handle very different types of information. The most efficient storage format for one type of information is different for other types of information. Another tactical variation is the variation among user interfaces. Some of the interfaces will be graphical while others will be text-oriented.

The tactical variations occur at places in core assets where product-specific information is required. The tactical variation in user interfaces results in a number of variation points. One such point might address the look and feel. Many windowing systems provide the ability to select different skins, menu styles, and window borders.

These variation points often correspond to variation points in the production method. For example, different products can require different binding times which in turn can require different design and construction techniques. Choosing one windowing systems could result in binding variation decisions at installation time while another windowing system would support configuration changes at any time.

4. Conclusions and Future Work

Variation modeling and production planning need to be cooperating activities. Production planning does not resolve product variations but they are vital to its success. At the very least the production method must provide the means for resolving all product variations. Frequently, product variations lead to variations in both the production method and production plan.

Variation in a software product line organization includes differences in the feature sets of products and in the system used to produce the products. These differences are captured in the assets related to product production. The production system must support these variations. That involves mapping the strategic variations into the core asset specifications including those of the production system itself.

The goals for a software product line impose goals and constraints on its production system. The production strategy, production method, and production plan provide a natural progression from strategic to tactical to concrete instructions for the product builder.

A number of issues require further investigation. Our techniques for identifying production variations are still intuitive and need more complete definition. The relationships between product and production variations and between strategic and tactical variations are not fully clarified. For example, it seems that each specific variation point has some element of both product and production associated with it. Is this true only for code-based assets or for assets in general?

There are multiple method engineering techniques that provide support for reusable pieces, i.e., method fragments. Which of these techniques can be used during method engineering to ensure that the fragments resulting from the different variants can be composed into a useful method? Will those techniques scale given the strategic levels of reuse encountered in a software product line organization?

Both production planning and variation management are central to product line success. The techniques presented in this paper illustrate that these two important ideas should work together to enhance product line operation.

5. Acknowledgments

Special permission to reproduce the Paper "Modeling Variation in Production Planning Artifacts" by Gary Chastek and John McGregor, Copyright 2008 by Carnegie Mellon University, is granted by the Software Engineering Institute.

6. References

[1] Clements, P. and L. Northrop, *Software Product Lines: Practices and Patterns*, Addison Wesley, Reading, MA (2002).

[2] Berg, K.; Bishop, J.; & Muthig, D.: Tracing Software Product Line Variability – From Problem to Solution Space, Proceedings of SAICSIT 2005, pp 182 –191.

[3] Tarr, P.; Ossher, H.; Harrison, W.; Sutton, S. M.: N Degrees of Separation: Multi-dimensional Separation of Concerns. ICSE'99, 1999, pp. 107 – 119.

[4] Schmid, K. and Eichelberger, H.: Model-Based Implementation of Meta-Variability Constructs: A Case Study using Aspects, http://www.vamos-workshop.net/2008/papers/VAMOS08_07.pdf

[5] Chastek, G.; Donohoe, P.; McGregor, J. D.: A Production System for Software Product Lines, SPLC 2007, pp. 117-128, 11th International Software Product Line Conference (SPLC 2007), 2007.

[6] Chastek, G.; Donohoe, P.; McGregor, J. D.: Formulation of a Production Strategy for a Software Product Line. Technical Note, CMU/SEI-2008-TN-023, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, USA, 2008.

[7] Chastek, G.; Donohoe, P.; McGregor, J. D.: Applying Goal-Driven Method Engineering to Product Production in a Software Product Line. Technical Report, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, USA, to appear.

[8] Chastek, G.; McGregor, J. D.: Guidelines for Developing a Product Line Production Plan, Technical Report, CMU/SEI-2002-TR-006, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, USA, June 2002.

[9] Porter, M. E., *Competitive Strategy*, Free Press, 2004.

[10] Brinkkemper, S. Method Engineering: Engineering of Information Systems Development Methods and Tools, Information and Software Technology 38 (1996) 275-280.