

# Early Aspects in Software Product Line in Product Production

Gary Chastek<sup>1</sup> and John D. McGregor<sup>2</sup>

<sup>1</sup> Software Engineering Institute, Carnegie Mellon University,  
Pittsburgh, PA 15213-3890, USA  
{gjc, pd}@sei.cmu.edu

<sup>2</sup> Clemson University, Clemson, South Carolina, 29634  
johnmc@cs.clemson.edu

**Abstract.** This position paper describes a problem in the development of a software product line that we believe can be solved using aspects. The core asset developer's and the product developer's perspectives on product variability differ. The core asset developer sees product variability locally, as it applies to their specific core asset. The product developer sees it globally, as it applies to all the core assets (i.e., across the product development). We believe that aspects are useful in coordinating these two views, and present an example in support of that belief.

## 1 Introduction

The fundamental goal of any software product line is to *produce products* in accordance with the business and market goals of an organization. Those goals apply not only to the products, but also to how those products are produced. Product production in a software product line as a system in its own right, and as such, the product line organization needs to plan for the success of its product production system. Otherwise, that software product line could fail because the product developers do not understand how to effectively build products from the core assets [1], or because those core assets have not been effectively coordinated for the production of products.

Clements [1] defines product production as consisting of a production strategy that describes how products will be produced, an “attached process” for every core asset that prescribes how that core asset is to be used to produce a product, and a production plan that documents how a product developer is to produce a product from the core assets (See Figure 1). Chastek [2] describes how the product line requirements relevant to product production are recognized and modeled, as a prequel to formation of a production strategy. Finally, Chastek [3] describes how the production strategy guides the development and coordination of the core assets, describes the relationship between the production strategy and plan, and details the structure and content of a production plan.

The purpose of this paper is to describe the potential role of early aspects in the production of products in a software product line.

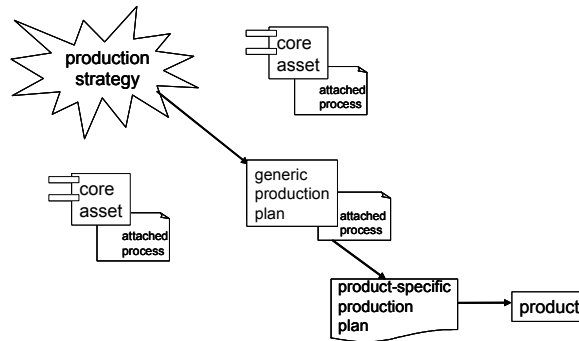


Fig. 1. Product Production in a Software Product Line

## 2 Product Production Concerns

The product production strategy defines concerns that conceptually cut across the definition of the core assets and their attached processes, as shown in Figure 2. Those concerns are derived from the product line goals that affect product production, such as “a faster time to market”.

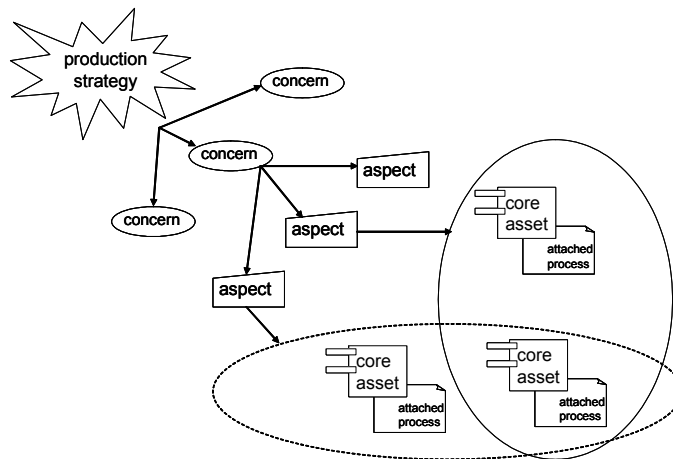


Fig. 2. Product Production Strategy Cutting Across Assets

The concerns cut across the core asset definitions because the primary decomposition of the core assets relates to how those core assets are built but not how products

are developed from them. For example, a particular core asset developer, the requirements engineer, models the requirements for all the products in the product line. Product variation points define those places in the requirements model where the requirements for various products differ. In the admittedly unrealistic example of a product line consisting of only two products (a fast- and a slow-responding product), the requirements model contains a variation point where it describes the response time, either fast or slow, of the products.

The product developer then has to build a specific product using the core assets. For the fast-responding product, that means that the product variation points corresponding to that product must be located in the requirements model and then instantiated to produce the fast-responding product requirements. The product developer has to repeat those “locate” and “instantiate” steps for all the core assets. The primary decomposition of the core assets forces the product developer to “locate and instantiate” the product variation points to create the needed corresponding piece of the product development. In other words, the primary decomposition of the core assets does not reflect how products are built.

Production concerns correspond to variation points that are defined at several points in the process of constructing a product line. They may be defined in the requirements definition, the software architecture, or the design. Any of these points can give rise to concerns. For example, a concern might be the degree of flexibility in the functionality of products. This concern would lead to consideration of the binding mechanism to use at some variation points in the product definition.

Each product production concern can be mapped on to multiple production aspects. Each aspect defines exactly how specific core assets are affected by that concern. For example, the degree of flexibility concern mentioned earlier may result in the use of mechanisms in the software architecture to support the late binding of functionality.

An aspect of the production strategy will affect many core assets of the product line and different aspects will affect different core assets. For example, one aspect might affect the structure of the software components and the content of their documentation. Another aspect might affect the steps in the build scripts and the execution of the automatically-generated test cases.

Consider an example in which the production strategy is for end-users to create applications out of the products in the product line. This strategy would produce an aspect related to naming conventions. Since the strategy calls for end-users to assemble their own applications, the aspect would support products having names that correspond to domain concepts that are familiar to the end-users. The production plan would have a product naming step and, in this case, instructions to select an appropriate domain concept as the name to be presented to the end-user programmers. See Figure 3.

### **3 Production Plans**

The core asset developers produce a generic production plan to accompany the production strategy. The plan provides those portions of the production process, sched-

ule, bill of materials and other artifacts that do not change from one product to another. The generic production plan is a template into which selected elements, such as product-specific schedules and a bill of materials are integrated to form a production plan that is specific to an individual product. The product-specific production plan contains, for example, a process definition that describes the complete product production process, as well as schedules, bill of materials, test cases, and any other artifact needed to create that specific product.

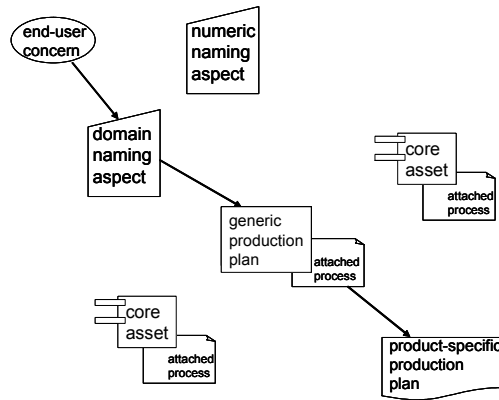


Fig. 3. Example Application of Aspects

The elements integrated into the product-specific production plan are production aspects. The goal of weaving these aspects into the generic production plan is ultimately to produce a product-specific production plan, see Figure 4. At this time the weaving is manual but method engineering techniques provide guidance on how we could automate portions of this task [4]. The production process in the product-specific production plan is formed by applying method engineering techniques to the attached processes of the core assets that are used in the production of the product.

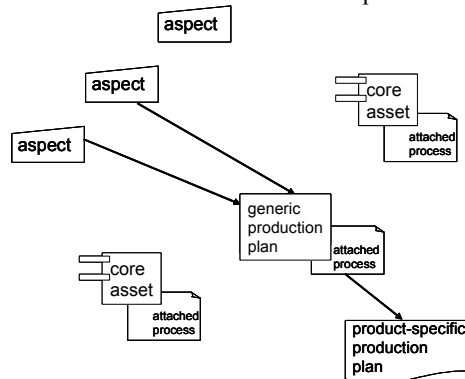


Fig. 4. Aspects Woven into a Generic Production Plan

## 4 Example: Production Concerns in an HIS

A home integration system (HIS) coordinates devices in the home, such as the lighting, stereos, TVs, heating and cooling, security systems, and offers the home owner centralized control over these devices. Two concerns can be identified in the production strategy for the HIS example presented in [5]. The production strategy in that example defines two steps: assembly and configuration. The configure production concern can be realized as a Plug'n Play aspect and as a manually-woven property file aspect. The core asset team decided that a Plug'n Play approach was the optimal strategy for achieving the configurable attribute of the product line. The Plug'n Play configure production aspect was introduced across the definitions of the core assets, their attached processes, and in the generic production plan, see Figure 5.

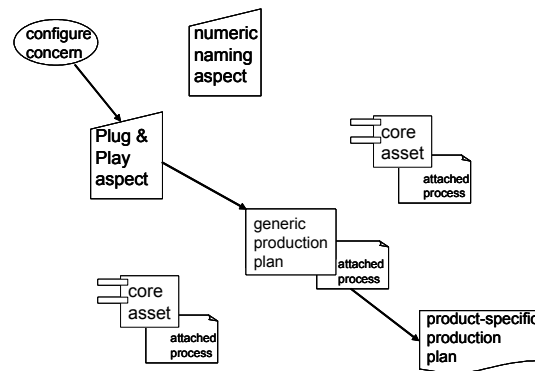


Fig. 5. The HIS Configure Aspect

The Plug'n Play aspect affects many of the core assets. The software architecture is defined to accommodate the signals that are raised when a unit, such as a new automatic coffee maker, is either inserted into or removed from the HIS. The user's manually-woven core asset will describe the Plug'n Play feature. The software components will be implemented to carry out the architectural vision of how Plug'n Play works. In each case, how the weaving is accomplished depends on the type of asset.

The production process is created by integrating the attached processes of the selected core assets. The attached processes are combined by analyzing the dependencies among the core assets. These may be input/output dependencies, shared resource dependencies or other dependencies. The outputs of one attached process may be mapped to the inputs of the next phase. The resulting process is used to build one specific product. In this case the second phase of product development will be carried out by the product itself.

## 5 Summary

We have presented a view of how concerns and aspects are being integrated into our work on product production in the context of a software product line. Our initial experience shows that this is a useful way to bring together the needs of the core asset builder and the product builder, to give each a view of the other's perspective, and to maintain consistencies between the two views. We intend to continue our work to determine the feasibility of automating as much of the product production planning as possible.

## 6 References

1. Clements, Paul and Linda Northrop, *Software Product Lines: Practices and Patterns*, Addison-Wesley, Boston, Massachusetts, 2002.
2. Chastek, Gary and Patrick Donohoe, *Product Line Analysis For Practitioners*, CMU/SEI-2003-TR-008, Software Engineering Institute, Pittsburgh, Pennsylvania, 2003.
3. Chastek, Gary and John D. McGregor. *Guidelines for Developing a Product Line Production Plan*, CMU/SEI-2002-TR-006, Software Engineering Institute, Pittsburgh, Pennsylvania, 2002.
4. Chastek, Gary, Patrick Donohoe, and John D. McGregor, *Applying Goal-Driven Method Engineering to Product Production in a Software Product Line*, CMU/SEI-2005-TN-034, Software Engineering Institute, Pittsburgh, Pennsylvania, 2005 (to appear).
5. Chastek, Gary, Patrick Donohoe, and John D. McGregor, *Product Line Production Planning for the Home Integration System Example*, CMU/SEI-2002-TN-029, Software Engineering Institute, Pittsburgh, Pennsylvania, 2002.