

# A Survey on Various Approaches used to manage Variability

**Ekta Yadav**

Department of Computer Science and Engineering  
ITMU  
Gurgaon, India  
ektay18@gmail.com

**Ms. Hitesh Yadav**

Department of Computer Science and Engineering  
ITMU  
Gurgaon, India  
hiteshyadav@itmindia.edu

*Abstract — In this era of high technology, a software needs to be able to reuse which comes through variability. Variant points are added at different levels of abstraction so that variability is introduced. This paper describes that how variability is introduced in software product lines and some modeling notation. Software product line is valuable for those organizations which has software product family. This paper targets to review various approaches to manage variability in software product line and in modeling.*

## I. INTRODUCTION

Variability is mushrooming day by day as there is a great demand of customized product in market. Variability of a Software or its component is the ability to change for different purposes in different environment. It can be studied mainly in every phase of software engineering [1]. On the other hand there is a different mechanism, Changeability is defined as a change which occur in a software over a period of time. Changeability is a planned activity not any change due to errors. There is a major difference between changeability and variability, variability can be done for customer needs but changeability is not for a customer specific issue. Variability is basically used in software product line engineering.

## II. VARIABILITY

Variability is the ability with which a system is subject to variation. It is very necessary to build a framework or SPL to reuse a software. A product also need to have variant points to reuse. It is very difficult to identify where variation points are needed? Is there any available variant points or where can we add new ones? And next very important question is who can add variant points – developer or reuser or both. All these things are very important for variability. There are certain factors like Time, Representation, Variants which help addressing these problems.

## III. SOFTWARE PRODUCT LINES

Let us consider an example of car factory which create different kinds of car. Obviously there are some standard parts and features which the company uses in almost all the cars that's where software product line is useful. Because it will be very time consuming and waste of resources to develop a product from scratch.

Software product line is the framework for the products which are related and they are supposed to use same technology, same methodology, same process and components [3].

It is impossible for system developers to provide customized product for all its customer and this customer requirement can't be ignore that's where software product line engineering provides a solution.

Different companies uses software product line engineering for their different products. Some of them are, Nokia for its mobile phones which produces 25-30 phones every year which supports multiple languages and platforms and also wide range of features. Hewlett Packard for its printer systems. Product line is comprised of core assets and applications. Core assets can be anything like architecture, documentation , specification, test cases and workplans etc.

### 1) Characteristics of SPL

- Product lines are large systems which comprised of thousands line of code or even more than that.
- All SPLs uses their own component models .These component models are very large and also they are not dependent on any other component.
- Every component of the SPL should have its own blackbox reuse .
- Changes in SPL improves quality of it.

#### IV. MODELLING VARIABILITY IN SOFTWARE PRODUCT LINE

##### A. Parameterization [2]

It provides access to a reuser. In this, a reuser can change values of a attribute of a component. It can be achieved through component interface in which user can either initialize or change value of attributes.

Parameterization is used in domain analysis method. We can take an example of FODA (Feature Oriented Domain Analysis) .FODA method helps in acquainting feature modelling to domain engineering.

Let us consider an example of ATM(Automated Teller machines). We will consider the design of banking system. It has six variation points to create a product line that can be used on several different banking solutions. This product line is helpful for the system which has several ATMs located at different geographical location and connected to a central network through WAN. Every device has its own cash dispenser, keyboard, display screen, card reader, and a receipt generator. Each ATM has a card reader, a cash dispenser, a keyboard display, and a receipt generator. There are six variation point which helps in start up and validation. Some example of variability in this case used are greeting, language, action. Greeting is used for display, to change the language of display, to change the action on card expiration. Let us take a example of ATM in which when a user transacts money from the system , it can display compliments saying “ thank you”, “ Have a nice day”, “good luck”. So it needs to set a parameter which is **compliment**, now a reuser can change value of the display parameter. Pictorial representation is shown in fig 1.

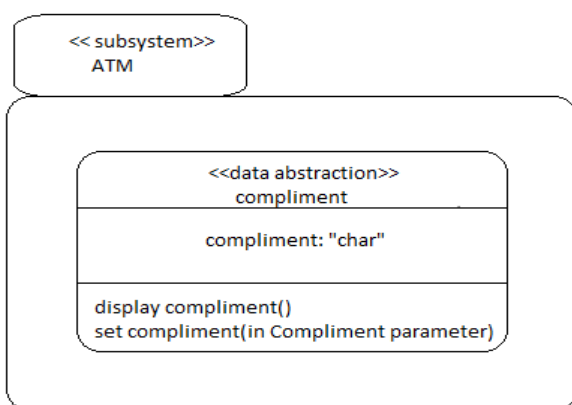


Fig.1 Compliment - Parameterization

##### B. Inheritance [2]

In this approach, variants are specialized version of the components. Instance of a super class is extended by a subclass it can either add new operation or override existing operations. Variants can use different interface unlike modeling variability using information hiding in which variants uses same interface. In this only limited number of variants available to a core asset component.

Let us consider the same example of banking system. There are various variation like greeting variation and language variation both are deal in a same way. A standard interface is defined for language variation in an abstract super class and also there is a subclass that is able to inherit some operations and can be implemented in any suitable language.

##### C. Information hiding [2]

In this approach, by using similar interface, various versions of a component can be created . There are different versions of same component which is called variant. Variability is defined in these versions. A reuser can select any one of the versions and can use it in application.

If we consider the example of banking system again , there are functionalities in which language and expiry of a card can be changed which can be implemented using information hiding. A class is used to design those variation which has the same interface. Classes used LanguagePromptEnglish and LanguagePromptSpanish realize the ILanguagePrompt interface.

#### V. MODELLING VARIABILITY USING MODELLING NOTATIONS

There are various notations used in variability like CBPMN ( Configurable Business Process Modelling Notations), CEPC(Configurable Event Process Chain).

##### A. CBPMN [5]

Business process modeling notation (BPMN) is modeling notation which is used to describe activity and process in sequence.

C-BPMN is a modeling language used to model configurability at activity level and decision level. This is one level higher than BPMN [5].

CBPMN basically deals with activities .Activities are any task which is going to be accomplished in business process. There are three kinds of activities- atomic activities , complex activities, configurable activity.

Atomic activities are those which cannot be decomposed into further activities .

Complex activity are those which can be broken into more than one atomic or configurable activities.

Configurable activities are those which can be individualized by preferring one instance [6].

There are decisions which support activities. Decisions uses gateways. Pictorial representations of configurable activities and decisions are shown in Fig.2.

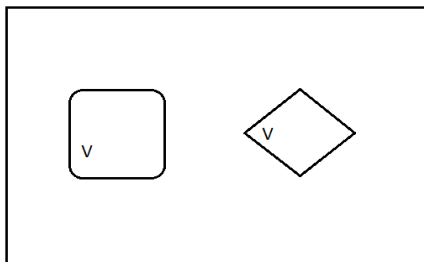


Fig.2 Configurable activity and decision

## References

- [1] Matthias Galster, Danny Weyns, Dan Tofan, Bartosz Michalik, and Paris Avgeriou, “Variability in Software Systems—A Systematic Literature Review”, IEEE transactions on software engineering, vol. 40, no. 3, march 2014.
- [2] Diana L. Webbera, Hassan Gomaab, “Modeling variability in software product lines with the variation point model”, ELSEVIER, April 2003.
- [3] Jilles van Gorp, Jan Bosch, Mikael Svahnberg, “On the notion of variability in software product lines”, IEEE ,Software Architecture,August 2001.
- [4] Jilles van Gorp, Jan Bosch, “Design , implementation and avolution of object oriented frameworks: concept and guidelines”,Software: Practice and Experience,Volume 31, Issue 3, pages 277–300, March 2001.
- [5] Deepak Kumar Sharma, Hitesh, Varun Rao , “Individualization of Process Model from Configurable Process Model constructed in C-BPMN” , IEEE, computing, Communication & Automation (ICCCA), 2015.
- [6] Deepak Kumar Sharma, Hitesh, Varun Rao, “ Configurable Business process Modelling Notation”, Advance Computing Conference (IACC), 2014 IEEE International.

“