IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 4 Issue 2, February 2017
ISSN (Online) 2348 – 7968 | Impact Factor (2016) – 5.264
www.ijiset.com

Tool Support for handling Variability in Class Diagrams

Ekta Yadav
ektay18@gmail.com

Hitesh Yadav
The NorthCap University
Gurgaon, India
hiteshi.vadav@gmail.com

Neeti Kashyap
The NorthCap University
Gurgaon, India
neetikashyap@ncuindia.edu

***Abstract* — In this era, where technology is changing rapidly there is strong need for software products that can adapt change which implies that software product should be reusable. For software reusability, some variant points are added in the product by the developer that's how variability is introduced. Variability is mainly required in software product line and it also imposes some challenges on software development. Software variability is aptness of a software product in which its components can be used for different purposes [1]. In this paper, Author is describing about notations and tool which is used to handle variability in class diagrams.**

I. INTRODUCTION

Variability is the mechanism in which a software product is made capable to be used for different purposes according to different user requirement[5]. It is very necessary to build a framework or SPL to reuse a software. An system likewise need variation focuses to reuse. It is extremely hard to distinguish where variety focuses are required[1]. During literature survey, some limitations are identified- major of them is lack of notations and tool support. There were no particular notations for variant points and variability approaches that makes vary difficult to locate variant points. To overcome that author provided some notations in the tool. A tool is also created in which class diagrams can be drawn and variant points can be indicated easily. Different approaches to identify variability is also explained with the help of tool.

II. VARIABILITY MODELLING NOTATIONS

There are various methods in which variability can be modeled [2]. Author is using some notations through which variant points can be easily accessed in UML diagrams. There are several methods Parameterization , Inheritance, Information hiding in Class diagrams .Table below represents different notations. These notations are depicted in following UML diagrams of Library management system .

III. DIFFERENT APPROACHES ARE DESCRIBED WITH THE HELP OF TOOL

Author is using UML notations to describe different approaches[7] . There are various tools/softwares available to model UML notations but like the problem statement says : Identification of methods to handle variability , the real problem lies in identifying variant points. Author is using Microsoft Visual Studio to create UML notations and to identify variant points. It has different graphical images like rectangle , line and also textbox which is used to make class diagrams.

NOTATIONS	APPROACHES
P	PARAMETERIZATION
I	INHERITANCE

Table 1. Notations

A. PARAMETERIZATION [2]

In parameterization , variability is outlined at the worth of the parameter. Parameterization is employed in domain analysis methodology. During this technique, totally different values can be passed to parameter by a reuser that has the authority over variant points, this makes system more dynamic. A reuser will pass multiple values through the interface. , Fig.1 shows a category BookPrice contains a methodsetBookPrice(in BookPriceParameter) that is employed to show totally different values of various books with respect to different buyers(school and students). So, this can be a variant point during this category.

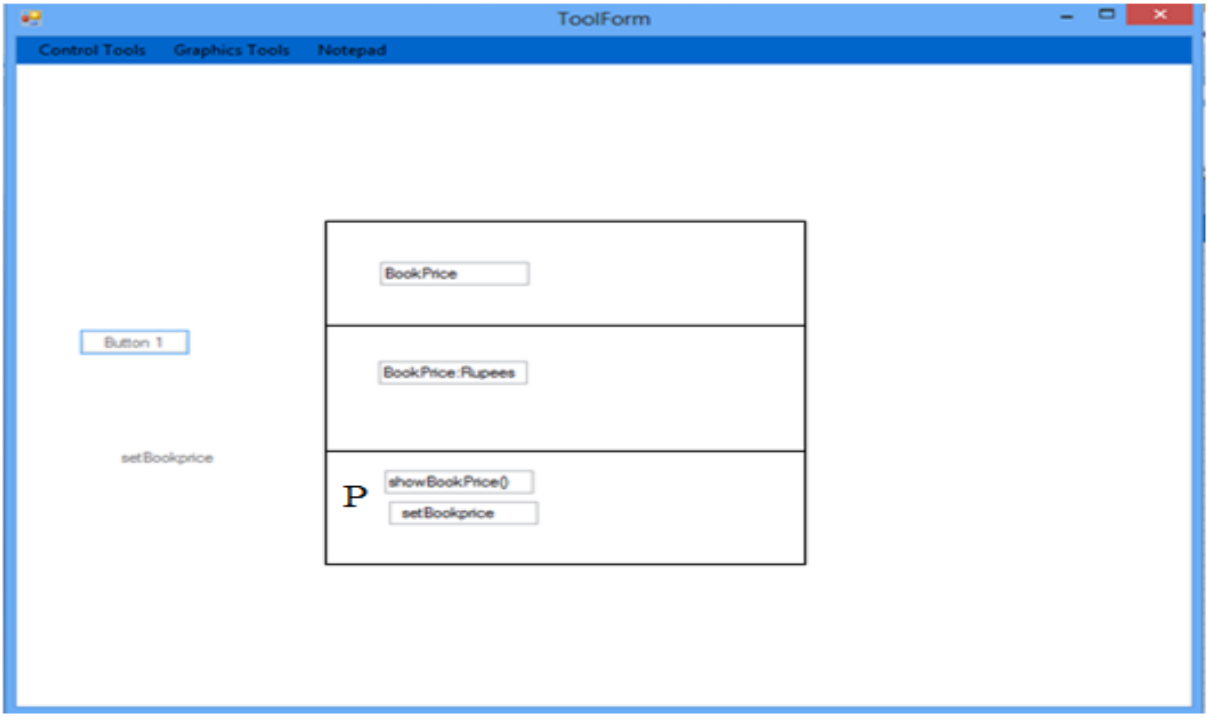

Figure 1 Snapshot of tool (Parameterization Approach)

B. INHERITANCE [2,5]

In this approach, 2 versions of a category doesn't would like to use same interface [2]. Super class has associate interface that is extended by the sub class, it will use already existing functions or use new operations [6]. Fig.2 shows an example of modeling variability through inheritance. It shows a super classs user that is extended by 2 subclasses, Faculty and PhdScholar. For each categories there's a subclass which can inherit functions from the super category interface. User has totally different information members like User_id,User_name that are heritable by subclasses, school andPhdScholar.

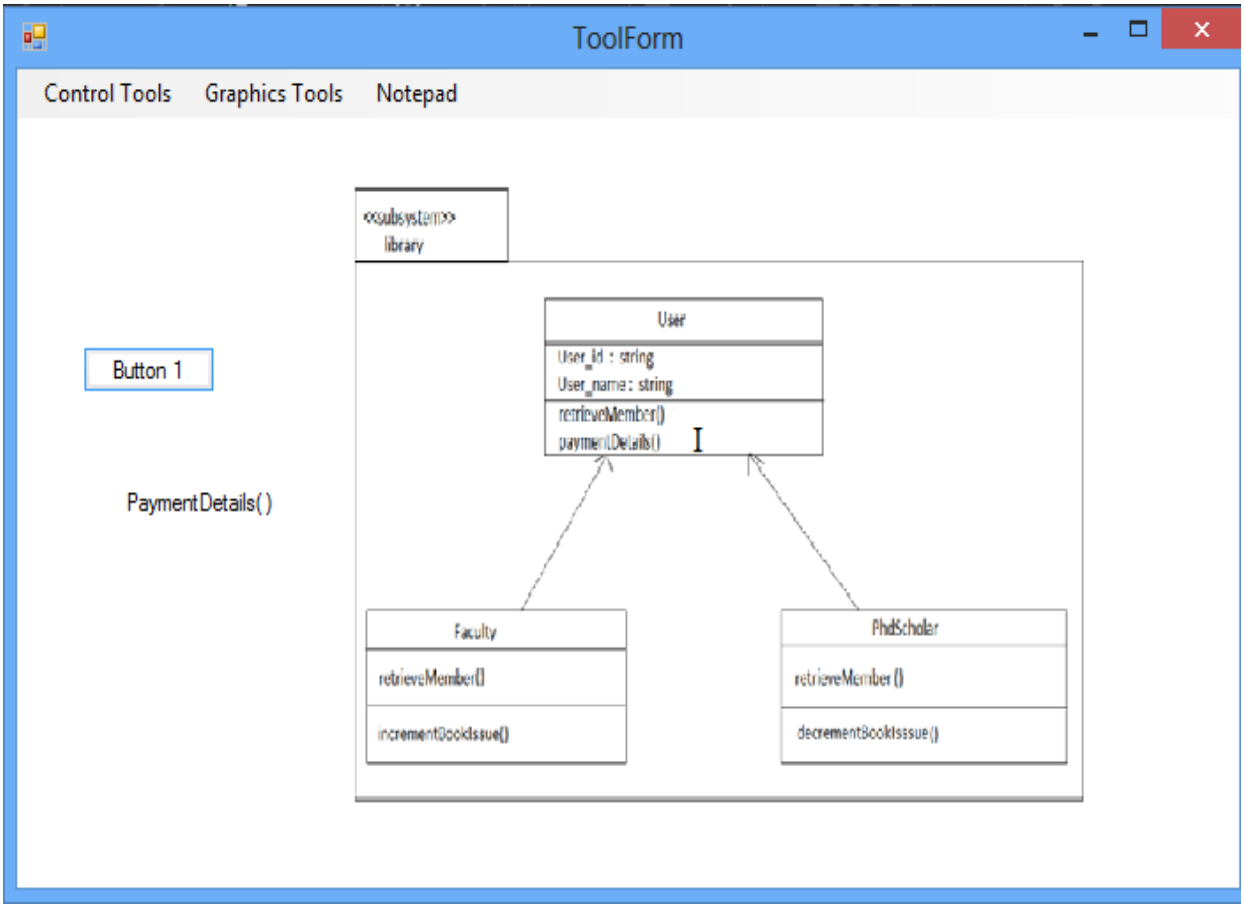

Figure 2 Snapshot of tool (inheritance approach)

C. INFORMATION HIDING [5]

In this methodology, by utilizing comparative interface, different versions of a segment can be made . There are diverse versions of same part which is called variant[2]. Variability is defined in these adaptations. A reuser can choose any of the versions and can utilize it in application. In data hiding variability is clouded in various rendition of the same interface. In this example, paymentFine () has two variant version payment card() and other one is payment cash(). Fig. 3 demonstrates this usage.

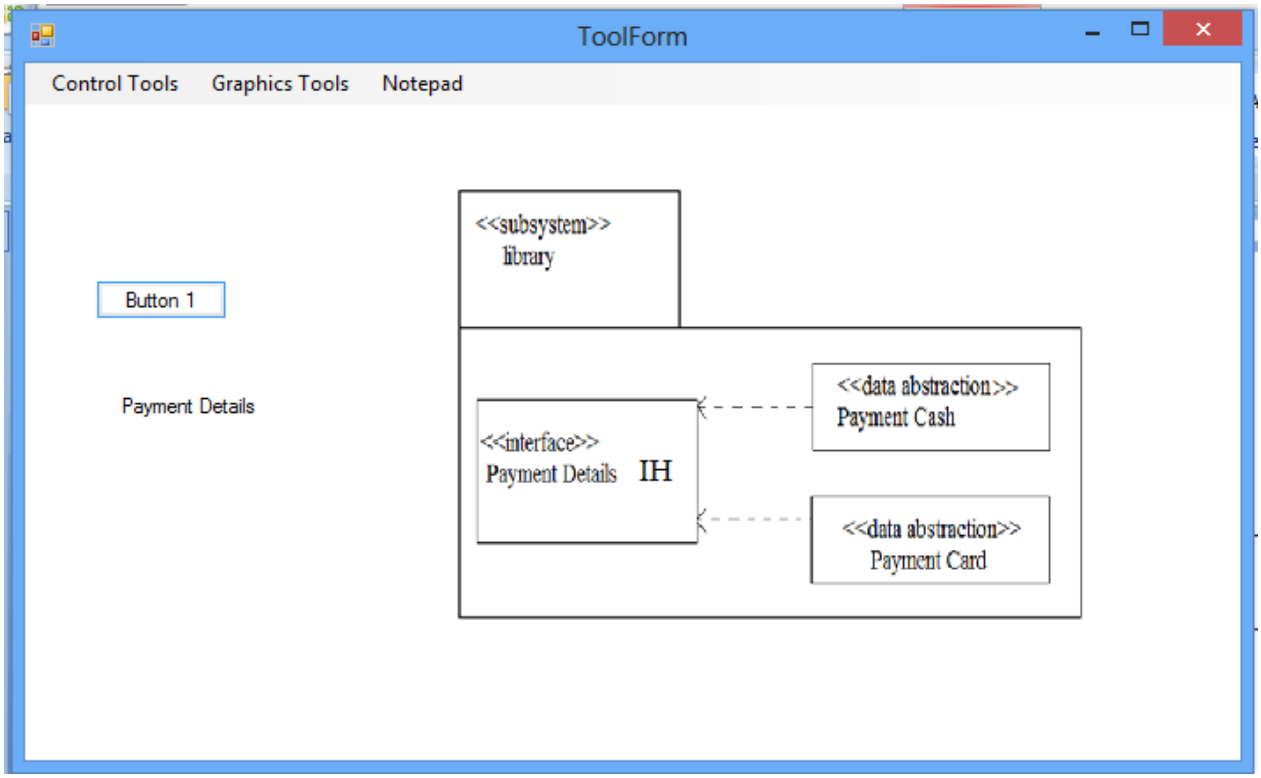

Figure 3 Snapshot of tool (information hiding approach)

IV. COMPARISION OF EXISTING TOOLS WITH NEW TOOL

Different tools are available for variability detection but each one has some limitations . They are compared with the new tool.

Table 2 .Comparision of existing tool with new too

	EXISTING TOOLS **BIZAGI, SPLOT, FODA VMC**	**NEW TOOL** **VARTOOL**
ROLE AND RESPONSIBILITY	**Used For Particular Uml Diagrams**	**Used For Class Diagrams And Activity Diagrams**
VARIABILITY NOTATIONS	**Variability Notations Are Not Available**	**Variability Notations Are Available**

REFERENCES

1 Matthias Galster, Danny Weyns, Dan Tofan, BartoszMichalik, and Paris Avgeriou, "Variability in Software Systems—ASystematic 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 Gurp, Jan Bosch, Mikael Svahnberg, "On the notion of variability in software product lines", IEEE,SoftwareArchitecture,August 2001.

4 Jilles van Gurp, Jan Bosch, "Design , implementation and avolution of object oriented frameworks: concept andguidelines",Software: Practice and Experience,Volume 31, Issue 3, pages 277–300, March 2001.

5 Hassan Gomaa, Diana L Webber, Modeling Adaptive and Evolvable Software Product Lines Using the Variation PointModel, Proceedings of the 37th Hawaii International Conference on System Sciences – 2004.

6 H. Gomaa, Designing Concurrent, Distributed, and Real-Time Applications with UML, Addison-Wesley, 2000.

7 G. Booch, J. Rumbaugh, I. Jacobson, The Unified ModelingLanguage User Guide, Addison-Wesley, Reading, MA, 1999.

8 D. K. Sharma, Hitesh and V. Rao, "Configurable Business Process Modeling Notation," *2014 IEEE International Advance Computing Conference (IACC)*, Gurgaon, 2014, pp. 1424-1429.

9 D. K. Sharma, Hitesh and V. Rao, "Individualization of process model from configurable process model constructed in C-BPMN," *International Conference on Computing, Communication & Automation*, Noida, 2015, pp. 750-754.

10 Ekta Yadav, Hitesh Yadav, “An Analysis on Various Approaches to Model Variability” International Journal of Innovative Science, Engineering & Technology, Vol. 3 Issue 5, May 2016.

11 Ekta Yadav, Hitesh Yadav, “A Survey on Various Approaches used to manage Variability” , International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 8, August 2015.