

PRODUCT LINE VARIABILITY MANAGEMENT USING TRACEABILITY INFORMATION

Rahila Ejaz, Naveed Ikram

Department of Computer Science, International Islamic University, Islamabad, Pakistan

Salma Imtiaz

Department of Computer Science, International Islamic University, Islamabad, Pakistan

Keywords: Variability Management, Product Line Engineering, Traceability.

Abstract: Variability management is an integral part of product line change management. The prerequisite for effective and efficient change management is traceability information. Traceability information supports understanding, maintenance and evolution of variability by establishing links between variability at various levels of abstraction and across development phases. Therefore an effective traceability based variability management is required for product line change management. But existing research on product line variability management does not explicitly state the traceability information for all the core issues of variability management which are variability identification, variability representation and realization, product instantiation and dependency management. This paper contributes by identifying a comprehensive list of variability management issues and traceability information related to these issues. We have proposed traceability based variability management model which maps the core issues of variability to the respective traces.

1 INTRODUCTION

Variability management is an essential element of product line change management which plays important role in successful software product line development (Van Gorp et al., 2001; de Oliveira et al., 2005; Buhne et al., 2005).

It is defined as *ability of a system or set of artifacts to be changed in a specific context* (Van Gorp et al., 2001). Bayer and Widden (2001) note that the need to manage variability increases as size of the product family increases. Variability management is crucial to satisfy the conflicting customer requirements. Variability management is also required when unanticipated variants emerge as a result of new requirements (Mohan and Ramesh, 2002), or missing variation points in architecture are realized later in the development lifecycle (Loesch and Ploedereder, 2007).

Traceability based variability management has been highly advocated in literature (Mohan and Ramesh, 2002; 2003; 2007; Van Gorp et al., 2000;

Metzger and Pohl, 2006; de Oliveira et al., 2005). It supports understanding, maintenance and evolution of variability by establishing links between variability at different levels of abstraction and across development phases. However variability management approaches in use tend to ignore the explicit traceability links for all the core issues of variability management.

We, in this paper have identified the issues of variability management and have mapped them to the respective traceability links in our proposed traceability model.

Section 2 describes concepts related to variability management, related work is discussed in section 3 and section 4 defines the proposed model. Conclusion and future work is presented in section 5.

2 CONCEPT OF VARIABILITY MANAGEMENT

Product proliferation, low cost and time to market are the main motives behind the family based development approach (Mohan and Ramesh, 2003), (Ajila et al, 2004). Proliferation in product variety is achievable by introducing variability into families of products to be developed (Vangurp et al, 2000;2001). Variability is a key concept of product line engineering which is usually set by customer specific requirements.(Ajila et al, 2004), (Deelstra et al, 2009).

Variability refers to behavioral difference and is made explicit through variant and variation points. Variation point is a location that represents the delayed design decision and variants are “set of values” that must be filled into variation point (de Oliveira et al, 2005), (Vangurp et al, 2001), (Bosch et al, 2001).

Researchers have identified that core issues of variability management are *Variability identification, Variability Representation, Variability Realization, Product instantiation and dependency Management*. These are called core issues because these are frequently reported in literature in the context of variability management (Bosch et al, 2001), (de Oliveira et al, 2005), (Mohan and Ramesh 2002; 2003; 2007), (Theil and Heindel, 2002), (Metzger and Pohl, 2006), (Lee and Muthig, 2006), (Estublier and Vega, 2005). From literature survey, we have further categorized these issues into sub issues as shown in figure 1.

<p>Variability identification Sources of Variation Evolution Aspects within a product Variation Between different product members</p>
<p>Variability Representation Requirements Level (feature) Design Level (design alternatives)</p>
<p>Variability Realization Variation point(VP),Variant(Var) Effects of variant on variation point Tasks attached to an individual variation point</p>
<p>Product Instantiation Application Specific Variability Systematic Reuse</p>
<p>Dependency Management Components Dependency Feature Tangling and Scattering</p>

Figure 1: Core issues of variability management.

3 RELATED WORK

A good deal of work has been done in the area of variability management with many people still working on it. Among them the most significant contribution is done by (Vangurp, 2000; 2001). Then (Svahnberg et al, 2000) and (Becker, 2003) are also among other contributors making variability management an understandable and un-ignorable concept for the product line engineering. (de Oliveira et al, 2005) also share his contribution in describing and improving the concept of variability management by proposing a UML based variability management process. Besides, there are several others who worked in the same lines. (Kannan, 2003), (Metzger and Pohl, 2006),(Estublier and Vega, 2005), (Buhne et al, 2005), (Deelstra et al, 2009).

The valuable contribution regarding traceability based variability management is done by Mohan and Ramesh (Mohan and Ramesh , 2002, 2003, 2007), (Jirapanthong and Zisman,2005),(Berg et al,2005) and (Kim et al, 2005).All of them emphasize on capturing various fragments of variability information like rationale of design decisions and possible alternatives used in later stages of product line development. Among them research work of Kim et al (2005) is the only one who explicitly states the traces for variability management however such set includes traces only for variability identification and realization and ignores other important issues of variability management that are reported in literature.(Becker, 2003), (Kim et al, 2005), and, (Estublier and Vega, 2005).For this reason, traceability based variability management model is proposed that incorporate explicit traceability information for all the core issues of variability management.

4 THE PROPOSED MODEL

In this section, we describe the three staged model proposed for variability management. Three stages of the model are issues identification, issues realization using traceability links and mapping between issues and traceability information.

1. Identify core issues of variability management
2. Issues realization using traceability information
3. Mapping between core issues and traceability information

Figure 2: Variability Management Model.

4.1 Identify Core Issues of Variability Management

In this section, we present the core issues (including sub issues) of variability management which are

- Variability identification
- Variability Representation
- Variability Realization
- Product Instantiation
- Dependency Management

4.1.1 Variability Identification

Variability identification is the first step in variability management which includes identification of variation points between products (de Oliveira et al, 2005),(Mohan and Ramesh, 2002).Early variability identification in the life cycle leads to better customization and improves company economy by increasing products variety effectively (Theil and Heindel, 2002).

Identification of variability includes identification of all causes of variability. Frequently reported causes of variations are

1. Change in market strategy, business needs or advances in technology (Ajila et al, 2004), (Faheem and Luis, 2007).
2. Stakeholders of the product line often have conflicting requirements which is a main source of variations, (Bayer. and Widden, 2001), (Ajila et al, 2004), thus these sources along with their requirements should also be identified.
3. Step wise extension of new requirements may introduce new variants (Riebisch and Plilippow, 2001).These variants should be identified as early as possible to avoid the architectural degeneration of product lines.
4. Evolution aspects within a product because it helps to decide about timely incorporation of a new member into product line (Jirapanthong and Zisman, 2005).

4.1.2 Variability Representation

The explicit representation of variability is proven to be essential in managing product line variability (Sinnema et al, 2004), (Jaring and Bosch, 2002). There are many ways to represent variability such as feature models, meta models, ontology of variability and UML notations. (Vangurp et al, 2000), (Metzger and Pohl, 2006), (Theil and Heindel, 2002),(Mohan and Ramesh ,2003).

Literature survey reveals that different modeling approaches have been proposed by authors to represent variability in different phases. E.g. feature modeling approaches are used to represent variability in problem space (Becker, 2003) whereas variability representation in design and architectural level is discussed by (Theil and Heindel, 2002), (Bachmann and Bass, 2001).

Jaring and Bosch (2002) indicate that although, variability representation is central to variability management, no standard notation is available and there is a lack of common frame of reference for variability representation.

4.1.3 Variability Realization

Variability realization is done at the implementation level. During variability realization, impact of variations on different software assets are understood and such variations are supported through appropriate implementation mechanisms (Becker, 2003).Variability realization includes variation points, their associated variants, effects of variants on variation points and tasks attached to an individual variant point (Kim et al, 2005).In other words, variability realization involves details of variability so that complex dependencies between different variations are comprehensible(Mohan and Ramesh,2003).

Literature suggests various implementation mechanisms to realize variability (Bachman and Bass, 2001).Becker (2003) emphasizes that variability realization is an important factor of variability management as it ensures variability implementation into product lines. Bosch et al, (2004) illustrates the importance of variability realization and concludes that without variability realization it is difficult to see the impacts of changes – which is essential for efficient and effective change management.

4.1.4 Product Instantiation

Products instantiation is a way to promote product variety for which reusability is known as a viable approach. However, sometimes products instantiation is not fully supported by reuse of core assets due to application specific variability.(Bayer, and Widden, 2001),(Kim, et al. 2005) which is important to manage in order to deliver products that are truly representative of customers' requirements but for product instantiation, reusability approach has two pitfalls which are" Identification of reusable components" and "selection of appropriate

configuration among the components “(Mohan and Ramesh, 2002, (Estublier and Vega, 2005).

Product line is successful if components presented in core assets are optimally used to develop the new product variants but literature highlights that complex dependency between components’ variation points, makes components configuration a tedious task (Theil and Heindel, 2002), (Mohan and Ramesh, 2002). Moreover, third party components are increasingly being used in product line based software engineering that also complicates the variability management process (Taulavuori, et al, 2004).

4.1.5 Dependency Management

Complexity because of complex interdependency between different artifacts of the product line is inherited attribute of product lines. Variation is always a part of some artifact therefore in order to manage variability, dependency management becomes critical. In addition to this, dependency management is important for variability management, as variability cannot be localized and it has widespread impact on product line artifacts (Becker, 2003). Dependency management is important due to

1. Components configuration because one variation point may be associated to more than one component, probably to be used in different context (Theil and Heindel, 2002).
2. Feature tangling and scattering (dependency between component and feature) is another issue of dependency management (Theil and Heindel, 2002), (Sinnema et al, 2004), (Loesch and Ploedereder, 2007) which creates problem during maintenance due to high rate of dependency.
3. Variability management requires consistent change integration throughout the software development lifecycle which is difficult due to *feature interaction* (Vangurp et al, 2000), (Lee and Muthig, 2006). Because features dependency is complex (Mohan and Ramesh, 2003) and impact analysis is difficult to perform in case of changed features.

4.2 Issues Realization using Traceability Information

Next step, after issues identification is to identify the traceability information for each issue. The purpose is to elicit the traceability links for each core issue

and its sub issues. These traceability links assist in providing a comprehensive overview about the type of artifacts and system elements required to tackle the different issues of variability management.

4.2.1 Traceability Information for Variability Identification

Discussed below are the types of variations that need to be identified.

4.2.1.1 Requirements Variability

(Vangurp et al., 2000; 2001) identifies that variability is generated in the form of requirements. It is then refined by feature diagram (Lee and Muthig, 2006) in the form of alternative, mandatory or optional feature (Bachmn and Bass, 2001), (Vangurp et al., 2001). This indicates that *from requirement to feature* is an appropriate trace for variability (functional) identification.

Product family has numerous members, each having its own set of requirements. Differences between them are inevitable which are utterly essential to identify in order to manage. One way to capture such differences is maintain traceability between artifacts (horizontal/vertical) of product members as discussed by (Jipanthrohg and Zisman, 2005). This trace helps to identify the differences between different product members by comparing the artifacts. In other words establish trace *between documents of product members (e.g. req to req, design to design, req to design etc.)* to identify the differences between product members.

4.2.1.2 Conflicting Quality Attributes requested by Stakeholders

Mohan and Ramesh, (2002) shows that it is important to trace the sources that demand conflicting quality attributes as it facilitates to justify the implementation of same components for different functionality. We can capture such information with the help of maintaining *from origin to conflicting quality attribute* trace. This trace is also helpful to identify all the sources of variations either internal or external.

4.2.1.3 Evolution Aspects within a Product

Jipanthrohg and Zisman, (2005) discusses that importance of identifying evolution aspects in a single product member. Such evolution is identifiable by maintaining the trace *from feature to architectural description to design documentation to*

code. This trace shows change incorporated in requirements and its effect on other artifacts. As a result, it provides a complete picture of change within a product.

4.2.2 Traceability Links for Variability Representation

It is clear that variability representation is not bounded to a single phase but it is an attribute of all phases of software development life cycle. It implies that this issue encompass artifacts of both problem and solution space. Berg et al, (2005) defines requirements related artifacts as part of problem space and architecture and implementation related artifacts in solution space. This indicates that variability representation involves following trace; *from requirements to design to implementation*. This trace defines variability of requirements in the form of features. Also it describes variability of architecture in terms of variation points and how this variability is then represented in design documentations and finally variability representation at code level; thus covering the whole domain i.e. problem and solution space.

4.2.3 Traceability Links for Variability Realization

In the context of variability realization, Kim, et al., (2005) has mentioned the artifacts and traces between artifacts. Commonality and variability specifications and Core asset model are stated artifacts for variability realization. Variability is specified in an abstract form by commonality and variability specifications which is then realized and refined by core asset model. In addition to this (Estublier and Vega, 2005) defines that *from feature to product line architecture* trace helps to realize variability at design time.

4.2.4 Traceability Links for Product Instantiation

4.2.4.1 Application Specific Variability

Product derivation is incomplete until application specific variability is not handled. To manage such variability we require to analyze the application specific features which are not supported by core assets. For this purpose Kim et, al, (2005) suggests application analysis model which includes details of application specific requirements. As a second step we need to identify the options and alternatives available to satisfy these requirements. For this

purpose, researchers suggest decision model (Berg et al, 2005), (Kim, et al, 2005), (Metzger and Pohl, 2006). Decision model contains details of alternatives and solutions in the form of variations and variation points. It implies that maintaining the trace *from application analysis model to decision resolution model* helps to map the application specific features to application specific variability and resolves the issue of application specific variability.

4.2.4.2 Identification of Reusable Components

Systematic reuse is proven to be an effective approach during product instantiation for which identification of reusable assets is critical. Jirapanthong and Zisman (2005) identified that maintaining the link *from product line architecture to product member* assists in identification of reusable components. Estublier and Vega, (2005) support this idea and defines that maintaining the *link between abstract product line architecture and reusable components* facilitates in extracting the functional components of product line which are then used for product instantiation.

4.2.4.3 Components Configuration

Reusable components should be configured appropriately to reap the full benefits of reuse. At the time of configuration, various choices are available and selection of right choice is essential to instantiate a right product. To do this, Bosch (1999) emphasizes to *maintain the alternatives and constraints that lead to configurability of various components*. Mohan and Ramesh, (2002) argued that maintaining such information also provides rationale for various architectural design decisions. We call this trace *from origin to architectural decision*. By origin we mean alternatives and constraints of configuration. We relate it to architectural decisions because reusable components are designed during architecture.

4.2.5 Traceability Links for Dependency Management

During dependency management three issues are reported (Theil and Heindel, 2002), (Loesch and Ploedereder, 2007), (Kathrin et al, 2005) (Becker, 2003)

1. Component dependency (between components)
2. Features tangling and scattering (between features and components)

3. Feature interaction (between features)

This implies that maintaining the trace *from component to component* addresses the issue of component dependency and *from feature to component* is required for the issue of feature scattering and tangling. For feature interaction we need *from feature to feature* and, *from feature to architectural decision to design documentation to implementation* trace.

4.3 Mapping between Core Issues of Variability Management and Traceability Information.

In this step, we present the mapping between the issues and related traceability links and summarize them in the form of table.

Table 1: Traceability links and variability management issues.

Traceability link	Issue
	Variability identification
From Requirement to feature.	Variability in Functionality
From Origin to Conflicting quality attributes	Conflicting quality attributes by customers
From Feature to architectural decision to design documentation to implementation	Evolution aspects within a product member
Relation between documents of same/different type for different product members	Variations among different product members
From requirements to design to implementation	Variability Representation
From Commonality and variability model to core assets	Variability Realization
From Feature to product line architecture	
	Product Instantiation
From application analysis model to decision resolution model	Application Specific Variability
From product line architecture to product member	Identification of reusable components
From abstract product line architecture to reusable component	
From origin to architectural decision	Components Configuration
	Dependency Management
From component to component	Component dependency
From feature to component	Feature Tangling and Scattering
From feature to feature	Feature Interaction

5 CONCLUSIONS AND FUTURE WORK

Variability management is an essential part during product line change management, contributing towards successful software product line development. Variability management is needed to ensure successful product lines in terms of efficient change management. Literature suggest traceability based variability management as an effective mean, however existing approaches do not explicitly state the traceability links for all the core issues of variability management. For the reason, the objective of this research was to propose a model that can manage variability whilst incorporating all the core issues of variability management.

Outcome of this research is the mapping between traceability information and core issues of variability management. We intend to validate the research by applying it in industry, to evaluate its effectiveness in the context of product line variability management. Incorporating the value based concepts in this model is another future direction.

REFERENCES

Mohan K. and Ramesh B, 2002 Managing Variability with Traceability in Product and Service Families. In *HICSS-35.02: Proceedings of the 35th Annual Hawaii International Conference on System Sciences*. Washington, DC, USA. IEEE Computer Society.

Mohan K. and Ramesh B, 2003 Ontology-based support for variability management In *HICSS-36.03: Proceedings of the 36th Hawaii International Conference on System Sciences*.

Mohan K. and Ramesh B , 2007. Tracing variations in software product families. *Communication. ACM 50, vol.12*. ACM Press.

Ajila, Samuel A., Kaba, Ali B., 2004. Using traceability mechanisms to support software product line evolution. In *IRI: Proceedings of IEEE International Conference on Information Reuse and Integration Las Vegas, Nevada, USA*.

Loesch.L, Ploedereeder,E. 2007. Optimization of Variability in Software Product Lines. In *SPLC: Proceedings of the International Software Product Line Conference*. IEEE Computer Society.

Van Gorp, et al. 2001. On the Notion of Variability in Software Product Lines'. In *WICSA': Proceedings of the Working IEEE/IFIP Conference on Software Architecture (WICSA'01)*, Washington, DC, USA. IEEE Computer Society.

Metzger,A., Pohl K., 2006. Variability Management in Software Product Line Engineering. In *Proceedings of*

- the 27th international conference on Software engineering.: ACM.
- Theil, S., & Heindel, A., 2002 Modeling and Using Product Line Variability in Automotive Systems. *IEEE Software. Special Issue on Software Product Lines*.19,4.
- Jirapanthong W., Zisman A., 2005. Supporting Product Line Development through Traceability. In *APSEC '05: Proceedings of the 12th Asia-Pacific Software engineering Conference*, Washington, DC, USA. IEEE Computer Society.
- Berg, K., et al., 2005. Tracing software product line variability: from problem to solution space. In *SAICT: Proceedings of the 2005 Annual Research Conference of the South African institute of Computer Scientists and information Technologists on IT Research in Developing Countries*. ACM International Conference Proceeding Series, vol. 150.
- Kim, D., S., et al., 2005 Traceability Map: Foundations to Automate for Product Line Engineering. In *SERA*. Proceedings of the Third ACIS int'L Conference on Software Engineering Research, Management and Applications. IEEE Computer Society, Washington, DC, 340-347.
- de Oliveira et al, 2005 A variability management process for software product lines. In *CASCON: Proceedings of the 2005 Conference of the Centre For Advanced Studies on Collaborative Research*. IBM Press.
- Taulavuori, A, et al. 2004. Component documentation-a key issue in software product lines'. *Information and Software Technology* 46(8).
- Becker, M. 2003 Towards a General Model of Variability in Product Families. in: *Proceedings of the 1st Workshop on Software Variability Management*, Groningen.
- Estublier, J. and Vega, G. 2005. Reuse and variability in large software applications. In *ESEC/FSE-13: Proceedings of the 10th European Software Engineering Conference Held Jointly with 13th ACM SIGSOFT international Symposium on Foundations of Software Engineering..* ACM, New York.
- Buhne, S., et al. 2005. Modeling requirements variability across product lines'. In *RE: 13th IEEE International Conference on Requirements Engineering*.
- Sinnema, M., et al, 2004. "COVAMOF": A Framework for Modeling Variability in Software Product Families, Proceedings of the Third Software Product Line Conference (SPLC), Springer Verlag Lecture Notes on Computer Science Vol. 3154 .
- Lee, J., & Muthig, D., 2006 Feature-oriented variability management in product line engineering. *Communications of the ACM*, 49(12).
- Bayer, J., & Widden, T., 2001 Introducing Traceability to Product Lines. LNCS 2290. In: Van der Linden, F. (Ed.), Springer-Verlag, Berlin.
- Riebisch, M., Plilippow, 2001 Evolution of Product Lines Using Traceability, OOPSLA Workshop on Engineering Complex Object-Oriented Systems for Evolution, Florida.
- Bachmann F, Bass L. 2001 Managing variability in software architectures. *Proceedings of the ACM Symposium on Software Reusability*.
- Jaring M, Bosch J. 2002 Representing variability in software product lines: A case study. *Second Product Line Conference (SPLC-2)*, (Lecture Notes in Computer Science, vol. 2379). Springer: Berlin.
- Deelstra, S., Sinnema, M., and Bosch, J. 2009. Variability assessment in software product families. *Published in information software technology Journal* 51(1).
- Bosch, J, et al, 2004 Variability issues in software product lines. In *PFE'4: Proceedings of International Workshop on Software Product Family Engineering*.
- Faheem, A., Luis Fernando, C. 2007, Managing the business of software product line : An empirical investigation of key business factors", *Information and Software Technology*, 49(2).
- Bosch, j. 1999. Product-Line Architectures in Industry: A Case Study. In *ICSE : Proceedings of the 20th International Conference on Software Engineering*.
- Svahnberg M, Bosch J. 2000. Issues concerning variability in software product lines. Proceedings of Third International Workshop on Software Architectures for Product Families. Springer, Las Palmas de Gran Canaria.