Representing Variants Including Quality Attributes

Clotilde Rohleder

Abstract. Developers must seriously address Non-Functional Requirements (Quality of Service) in the production of software families that include variants for different customers. Most prior research in this area deals with design and implementation aspects such as mechanisms that help implement the variability in software architecture. Few researchers have addressed how to represent variability in Non-Functional Requirements. This paper proposes a goal driven approach that captures the variability at both Functional and Non-Functional Requirements level. We use a goal driven formalism to represent the feature variability including the quality attributes through relationships. Our approach provides a global view of variants having different quality attributes and facilitates matching between the requirements and the product. It exposes the user to the choices that are relevant to the satisfaction of user goals. To identify the impact of Non-Functional Requirements on variants, we represent the Non-Functional Requirements by goals according to several decomposition methods. We capture the variability through requirements analysis and represent the variants through a goal-driven modeling formalism called "map." Each variant has its own quality attributes.

Keywords: Software variability, Quality of Service, Non-Functional Requirements.

1 Introduction

Developing any system, even one for a single customer, requires addressing the customer's Functional and Non-Functional Requirements (Quality of Service). Unfortunately, as mentioned in [4] and [5], most prior researchers have neglected the representation of the variability of requirements and have not addressed the impact of Non-Functional Requirements on variants [2], [3], [9], [10], [15].

We propose treating the variability from a Functional and Non-Functional Requirements perspective. To identify the impact of Non-Functional Requirements on variants, we propose to represent the Non-Functional Requirements by goals [6], [7], [8],[14] according to the decomposition methods of [1], [6], [7], [8] and [14], and to capture the variability through requirements analysis and to represent the variants through a goal-driven modeling formalism called "map" [12], [13], [4] and [5].

Centre de Recherche en Informatique, 90 rue de Tolbiac, F-75013 Paris, France
SIEMENS PL (DE) GmbH, Lina-Ammon-Strasse 22
D-90471 Nuremberg, Germany
fraipont@univ-paris1.fr, clotilde.rohleder@siemens.com

Each identified variant will have its own quality attributes.

The remainder of this paper is structured as follows. Section 2 introduces the Quality Variation Model, including the variant formalism using map, and the Non-Functional representation using NFR decomposition. This section shows with an example how to represent variability including Non-Functional Requirements. Finally, we summarize our work and conclude with plans for future work

2 Quality Variation Model QVaM

2.1 Overview of our model

In moving to the target systems, we consider design techniques of map to achieve Functional Requirements represented as goals and strategies. Functional Requirements are represented as variants. A variant is a representation at requirements level of a cohesive bundle of system functionalities according the user's point of view. In the variant representation, we include the Non-Functional Requirements and translate the impact of Non-Functional Requirements on Variants as quality attributes. During variants selection one can apply the impact rules of the represented Non-Functional Requirements on variants. This variants selection shows consequences at design level by the selection, implementation or configuration of system functionalities

2.2 Variants

Variants are based on the map model [12], [13], [4], [5]. Map is a process model expressed in a goal driven perspective. It provides a system representation based on a non-deterministic ordering of goals and strategies. Map features have four kinds of relationships, namely *multi-thread*, *bundle*, *path* and *multi-path*. These relationships show the possible combination of features from which the user can select the appropriate ones according to user needs. We map these combinations of features to variants. A **variant** is a representation at requirements level of a cohesive bundle of system functionalities according the user's point of view. We define different variant types corresponding to the different relationship types inside the map: *atomic*, *simple* and *composite* variant.

Textual notation of variants

We describe variants with codes (Table 1). Additionally we need the variant name, the

Source goal, the Target goal, the Manner and Application Rules [4], [5]. For all variant types, the *name* of the multi-path composite variant is the target goal. The *source and target goals* are source goal (which has the code *a*) and a target goal (which has the code *b*). The *manner* is expressed by a strategic path. Table 1 shows all variant codes and strategic path (*Q* is the intermediate goals bundle).

Variant types	Variant Code	Strategic path
Atomic Variant	ab_k	ab_k
Simple Variant with Alternate Choice	SVa_{ab}	$\otimes(ab_1, ab_2, .., ab_n)$
Simple Variant with Multiple Choice	SVm_{ab}	$\vee(ab_1, ab_2, ..., ab_n)$
Path Composite Variant	$CVp_{a,Q,b}$	$.(V_i, \ldots, V_n)$
Multi-Path Composite Variant	$CVm_{a,Qi, \ldots, Qn,b}$	$\cup(V_i, \ldots V_n)$

Table 1. Textual representation of variants

2.3 Integration of Non-Functional Requirements

NFRs are rarely "satisfied" in a particular clear-cut sense [6], [7]. Instead they affect decisions to contribute to, or hinder that a particular goal. Therefore, we used goals satisficing to suggest that generated software is expected to satisfice NFRs within acceptable limits, rather than absolutely. To concretely analyze and understand the impact of each NFR on variants, we have to decompose the NFRs into quality softgoals [1], [6], [7], [8] and [14]. Figure 1 shows the graphical and textual representation of Non-Functional Requirements. Graphically the NFR is represented by a circle. The NFR circle is named by the identified NFR and its decomposed NFR goal within brackets. For the textual representation of the NFRs goals, we need a *code,* a *name,* a *subject goal, satisficing data,* and a *coefficient*.

Graphical representation	Textual representation
NFR Code [NFR subject goal]	Code: <NFR Code> Name: <NFR name> Subject Goal: <NFR subject goal> Satisficing Data: <absolute satisficing result, relative satisficing result> Coefficient: <Importance weight of NFR>

Figure 1. Representation Non-Functional Requirements

2.4 Representation of NFR Impact on different types of variant

After having captured, defined and represented the variants and the NFRs, we research the impact of NFRs on variants. To consider the impact of Non-functional Requirements on variants, we use a catalogue of interrelationships that describe contributions of Non-Functional Requirements toward meeting goals/variants. We use satisficing links whose five values are recorded in Table 2.

NFR Impact on Variants	Very positive impact	Positive impact	Neutral impact	Negative impact	Very negative impact
Symbol	**++**	**+**	?	-	--

Table 2. Different NFR impact values on variants.

2.4.1 Atomic variants are not decomposable into other variants. They are linked directly to system functionalities. An atomic variant describes how to reach *directly* the target situation (concretized by the target goal satisfaction) from a initial situation (that has been reached after the source goal has been realized). The atomic variants are linked with each other to build variants with bigger granularity (simple or composite).

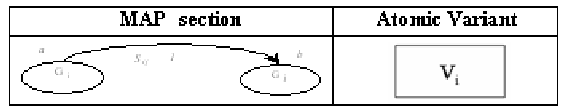

Figure 2 Example of map section and representation of its corresponding Atomic Variant.

These links associate the NFRs with variants and they describe which NFR gives which impact on which variant. The information concerning the NFR impact on each variant will be considered as a quality attribute for this variant. We will first deal with the graphical representation and then the textual representation of NFR impact on variants. The representation of NFRs'impact on variants requires satisficing links. We start the satisficing link from the NFR decomposed goal to the variant represented by a circle. The end of the link does not touch the variant circle to avoid confusion with the decomposition process explained by [6], [7], [8] and [14]. The link is completed by the satisficing NFR impact results. **Figure 3** is an example of representing some Atomic Variants with quality attributes.

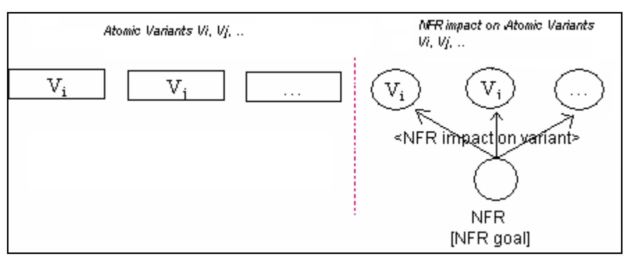

Figure 3. NFR Impact on Atomic Variants

2.4.2 Simple variants represent requirement variability by grouping the atomic variants that are linked either by an *alternate choice link* (Simple Variant with Alternate Choice) or by a multiple choice link (Simple Variant with Multiple Choice). In the first case, the atomic variants are mutually exclusive. This link expresses an exclusive choice between all atomic variants. Only one variant can be selected among several. Each atomic variant represents a manner or a distinct strategy in order to reach the variant target goal from its source goal. In the second case at least one atomic variant must be selected. The atomic variants are complementary. The satisfaction of target goal is done through the selection of one or more among those variants.

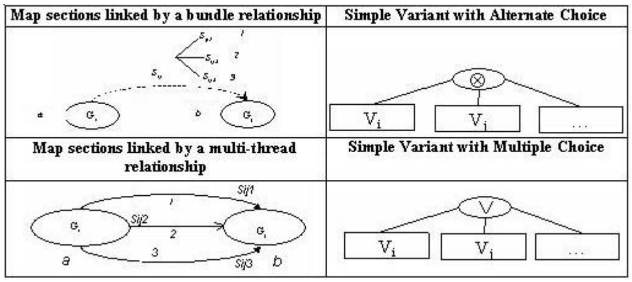

Figure 4. Map sections linked by a bundle or multi-thread relationship and Simple Variant with Alternate or Multiple Choice.

In order to implement the NFR Impact of Simple and/or Composite Variants without losing the understandability of the graphical representation, we have developed a NFR impact hierarchy. The main architecture principles are defined as follows. The NFR impact architecture implements the different variation types in assigning a NFR impact layer to each variant of our model. In choosing to perform the NFR impact for each atomic variant separately, we can then get the NFR impact of a bundle of atomic variants that are linked by an alternative or a multiple choice, on a higher layer. **Figure** 5 is an example of graphical representation of Simple Variants including quality attributes.

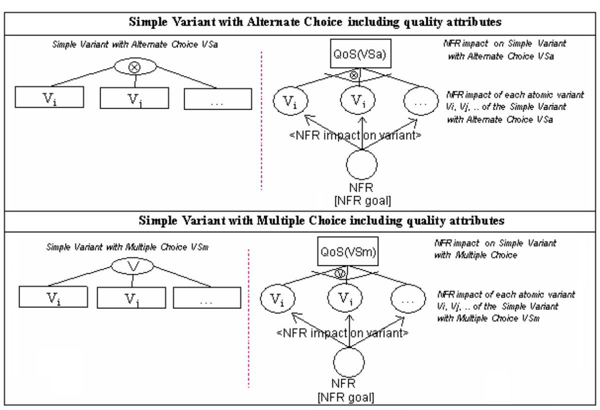

Figure 5. Simple Variants including quality attributes

2.4.3 Composite Variant

The Path Composite Variant consists of a *simple composition link* which links atomic variants, simple variants and/or composite variant, under a plan form which defines the order in which the variants must be realized. In a general way, a path composite variant is grouping all possible variants combinations between a source and a target goal through the satisfaction of an intermediate goals bundle. Each combination goes through the same intermediate goals bundle. The variations are in the manners that lead to satisfy each goal of the intermediate goals bundle. A Multi-Path Composite Variant expresses a variation in the selection of the intermediate goals which lead to satisfy the target goal from a source goal. Each possible combination of intermediate goal builds a distinct way. The satisfaction of the target goal implies the selection of distinct intermediate goals. Structurally, a Multi-Path Composite Variant (Figure 6) consists of a *multiple composition link* between variants. The Multi-Path Composite Variant must consist of at least one Composite or Simple Variant. Each sub variant is a possible variant combination which constitutes a possible way between source and target goal.

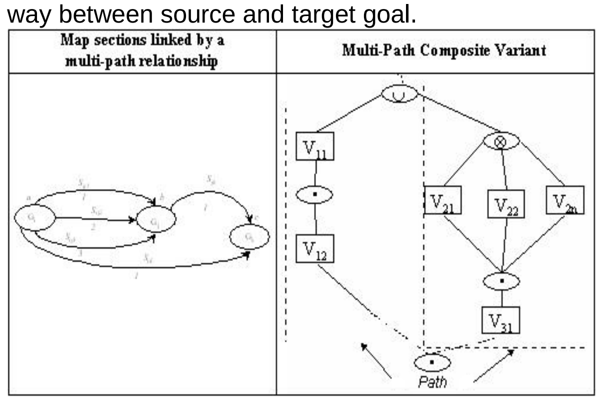

Figure 6. Map sections linked by a multi-path relationship and Multi-Path Composite Variant

In applying the same approach with variants which are linked with each other by a simple composition link, we get the NFR Impact of Path and Multi-Path Composite Variant in setting the NFR impact of each path. The NFR impact of the root variant is computed by the NFR impact of the sub variants. The NFR impact of the sub variant can be self computed by another NFR impact if the variant is self composed by sub-paths. So, the representation of the NFR impact of a composite variant consists of a hierarchy of NFR impacts which are linked by composition. **Figure** 7 represents the NFR impact of a Composite Variants.

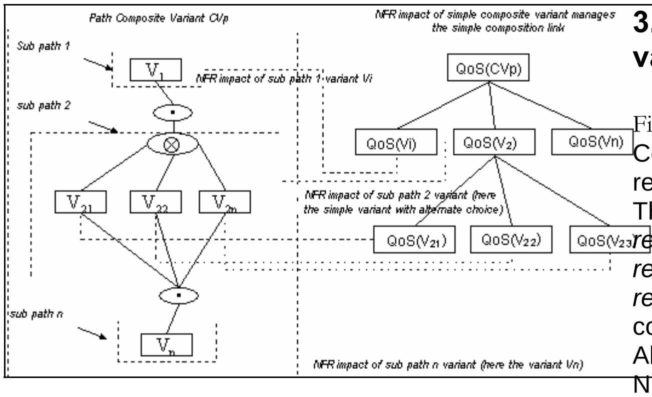

Figure 7. Composite Variants including quality attributes

The textual representation of NFR Impact on Variants completes the textual representation of variants [4], [5] using the NFR dimension that is considered as a quality attribute for this variant. The notation of the quality attribute inside the textual representation of variant Vi is QoS(Vi), which means Quality of Service of variant Vi. The quality attributes of the variant Vi is written as follows:

QoS(<Vi>*)*=
<Vi>.NFR1[$goal_{NFR1}$].<ImpactValue>…<Vi>.NFR_n .[$goal_{NFRn}$].<ImpactValue>

3. Example of representing the variants with quality attributes

Figure 8 is the representation of the Path Composite Variant $CVp_{a\{b\}c}$. This variant represents the cancellation of a paid reservation. The letter *a* is the code of the goal *To make a reservation*, *b* is the code of goal *To pay for a reservation* and *c* is the code of goal *To cancel a reservation*. SVa_{ab} and bc_1 are the corresponding codes of the Simple Variant with Alternate Choice and the Atomic Variant. The NFR Impact of $CVp_{a\{b\}c}$ will consider the quality attribute of SVa_{ab} and bc_1. The NFR impact on atomic variants ab_1, ab_2 and ab_3, which build the quality attribute of Simple Variant with Alternate Choice SVa_{ab} are represented through circles linked by the symbol of alternate choice link *«⊗».* The NFR impact of atomic variant bc_1 is attached to the other NFR impact by the sequence link symbol «.».

For example, the quality attribute of Atomic Variant ab_1 is QoS(ab_l) = ab_1.Performance[PerfGoal].++, ab_1.Security[SecurGoal].++. The quality attribute of Simple Variant with Alternate Choice SVa_{ab} is QoS(SVa_{ab})) ⊗ [QoS(ab_l), QoS(ab_2), QoS(ab_3)]. The quality attribute of Path Composite Variant $CVp_{a\{b\}c}$ is QoS($CVp_{a\{b\}c}$) = .[(QoS(SVa_{ab})), (bc_1.Informativeness[InformGoal].+)].

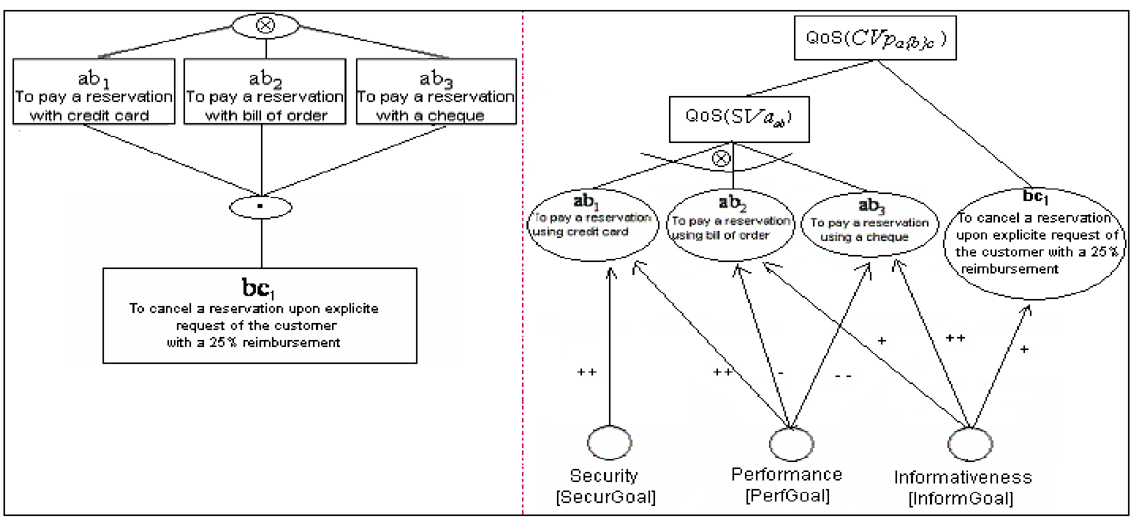

Figure 8. Graphical representation of NFR impact on Path Composite Variant $CVp_{a\{b\}c}$

4 Conclusion

This paper proposes a new way to express quality feature variability. We use a goal driven formalism to represent the feature variability including the quality attributes through relationships. Through our approach, the customer gets a global view of variants having different quality attributes without being lost in technical details. A representation of the variants at the Functional and Non-Functional Requirements level facilitates the matching between his requirements and the product. However, our proposal is a work in progress. In further work we will consider the task of building the correct derived product for different companies.

References

[1] Castro, J., Kolp, M., Mylopoulos, J. "Towards Requirements-Driven Software Development Methodology: The Tropos Project," *Information Systems*, June 2002

[2] Bachmann F., Bass L. "Managing variability in software architecture". *ACM Press*, NY, USA, 2001

[3] Bosch J., Florijn G., Greefhorst D., Kuusela J., Obbink H., Pohl K. "Variability issues in Software Product Lines." *4th International Workshop on Product Family Engineering (PEE-4)*, Bilbao, Spain, 2001

[4] Bennasri S., Souveyet C. "Capturing requirements variability into components", *6th International Conference on Enterprise Information Systems (ICEIS'04)*, Porto, Portugal, 2004

[5] Bennasri S. "Une approche intentionnelle de representation et de réalisation de la variabilité dans un système logiciel", *PhD dissertation*, Université Paris 1, 2005

[6] Chung, L., Nixon, B.A., Yu, E. and Mylopoulos, J. "Non-Functional Requirements in Software Engineering", *Kluwer Academic Publishers*, Boston, 2000

[7] Chung L., Nixon B. A., Yu E., Dealing with Change : An approach Using Non-functional Requirements, *Proceedings of the Second Internation Symposium on Requirements Engineering*, York, England, Springer Verlag London Limited, Requirements Engineering Journal, 1996, p. 238-260

[8] Chung, L., and Subramanian, N. "Process-Oriented Metrics for Software Architecture Adaptability", *Proceedings of ISRE*, 2001

[9] González-Baixauli B., Sampaio do Prado Leite J.C., Mylopoulos, J. "Visual Variability Analysis for Goal Models". *Requirements Engineering Conference* 2004: 198-207

[10] Halmans, Pohl K. "Communicating the variability of a software product family to customers", *Software and System Modeling*, Springer-Verlag 2003

[11] Keller, R.K., Schauer, R. "Design Components: Towards Software Composition at the Design Level", *Proceedings of International Conference on Software Engineering*, April 19-25, 1998, Kyoto, Japan, pp. 302-311

[12] Rolland, Prakash N. "On the Adequate Modeling of Business Process Families", *BPMDS'07 in conjunction with CAiSE'07,* Norway, 2007

[13] Rolland C. "Bridging the gap between Organizational needs and ERP functionality". *RE Journal*, 2000

[14] Subramanian N., Chung L. "Relationship between the Whole of Software Architecture and its Parts: An NFR Perspective"*. SNPD* 2005: 164-169

[15] Svahnberg, M., Van Gurp J., Bosch J. "On the notion of variability in Software Product Lines". *Proceedings of the Working IEEE/IFIP Conference on Software architecture*, 2001