

Zentrale Variabilitätsmodellierung für Requirements Artefakte in der Produktlinienentwicklung

Stan Bühne, Günter Halmans,
Kim Lauenroth, Klaus Pohl
Software Systems Engineering
Institute for Computer Science and Business Information Systems (ICB)
University of Duisburg-Essen
45117 Essen; Germany
(buehne/halmans/lauenroth/pohl)@sse.uni-essen.de

1 Einleitung

Software-Produktlinien haben das Ziel, der Schnelligkeit der Märkte und immer höher werdenden Qualitätsansprüchen durch Wiederverwendung von Artefakten gerecht zu werden [CN01]. Wieder verwendbare Artefakte umfassen alle Teilprodukte der Entwicklung, z.B. Anforderungsspezifikationen, Modelle, Code-Komponenten oder Testfälle. Die Entwicklung von Software-Produktlinien (im Folgenden mit Produktlinie bezeichnet) basiert auf zwei wesentlichen Konzepten: Variabilität und Trennung des Entwicklungsprozesses in Domain Engineering und Application Engineering.

Die Variabilität einer Produktlinie ermöglicht die Entwicklung von Produkten mit einem gemeinsamen und einem individuellen Teil. Durch die Ausnutzung von Produktlinienvariabilität, d.h. der Auswahl von Varianten bezüglich definierter Variationspunkte, lassen sich unterschiedliche Produkte entwickeln. Im Domain Engineering (Konstruktionsprozess) werden die allgemeingültigen Produktfamilien-Artefakte und die Produktlinienvariabilität definiert. Im Application Engineering (Produktionsprozess) werden auf der Basis der realisierten Produktlinien-Artefakte Produkte definiert und realisiert. [Va02].

Wir betrachten im Folgenden das Requirements Engineering in der Produktlinienentwicklung. Folgende Herausforderungen stellen sich hier durch die genannten Konzepte gegenüber der Einzelproduktentwicklung:

- Im Domain Engineering müssen Gemeinsamkeiten und Variationsmöglichkeiten (inklusive Abhängigkeiten) definiert und dokumentiert werden.
- Im Application Engineering muss die Produktlinienvariabilität kommuniziert werden, damit z.B. Kunden die Möglichkeiten der Produktlinie erkennen.
- Die Kommunikation von Produktlinienvariabilität erfordert eine geeignete Repräsentation dieser Variabilität [HP03].
- Die Repräsentation der Variabilität muss unterschiedliche Anforderungsartefakte berücksichtigen,

damit Sie unterschiedlichen Sichten von Stakeholdern gerecht werden kann. (z.B. Ziele, Szenarien oder modelbasierte Anforderungen).

Ein Lösungsansatz, die genannten Herausforderungen zu bewältigen, liegt zum einen in einer zentralen Variabilitätsmodellierung und zum anderen in der Nutzung der Beziehungen zwischen den Anforderungsartefakten.

2 Modellierung von Variabilität

Mit der Modellierung von Produktlinienvariabilität werden die Variationsmöglichkeiten einer Produktlinie definiert und dokumentiert. Variabilität wird mit Hilfe von Variationspunkten, den zugehörigen Varianten und den Beziehungen zwischen Varianten und Variationspunkten modelliert. [BGL03]. Wir unterscheiden zwei Arten der Modellierung von Produktlinienvariabilität:

- *Integrierte Modellierung*: Die Variabilität wird in das jeweilige Anforderungsmodell integriert. Beispielsweise werden in einem Use Case Modell Variationspunkte, Varianten und ihre Beziehungen eingeführt [HP03].
- *Zentrale Modellierung*: Die Variabilität einer Produktlinie wird in einem eigenen Modell definiert und eine Beziehung der modellierten Varianten zu den Anforderungsartefakten hergestellt [BGL03].

Die Repräsentation von Variabilität in einem zentralen Modell ermöglicht eine durchgängige Nachvollziehbarkeit bezüglich der Variationspunkte, Varianten und ihren Abhängigkeiten untereinander über alle Entwicklungsphasen hinweg. Darüber hinaus müssen die bestehenden Anforderungsmodelle (z.B. Use Case Modell) im Gegensatz zur integrierten Variabilitätsmodellierung nicht um die Repräsentationsaspekte der Variabilität erweitert werden. Eine ausführliche Erläuterung des zugrunde liegenden Metamodells für zentrale Variabilitätsmodellierung findet sich in [BLP04].

3 Produktlinienvariabilität in Zielen, Szenarien und modellbasierten Anforderungen

Das Requirements Engineering im Domain und Application Engineering gewinnt und dokumentiert unterschiedliche Arten von Anforderungen (wie auch die Einzelproduktentwicklung). Wir unterscheiden drei Arten:

- *Zielorientierte Modelle* dokumentieren die mit dem betrachteten Produkt verbundene Vision und verfeinern diese Vision durch weitere Unterziele. Sie beschreiben den Mehrwert eines Produktes.
- *Szenarien* beschreiben konkrete Anwendungsfälle. Sie stellen konkrete Benutzer/System-Interaktionen dar, die zur Erfüllung bzw. Nichterfüllung eines Zieles führen.
- *Modellbasierte Anforderungen*, z.B. ER-Diagramme, beinhalten oft die Beschreibung einer prinzipiellen Lösung wie bspw. grobe Datenstrukturen oder das Verhalten von Objekten.

Die drei Arten von Anforderungen stehen untereinander in Beziehung. Die Definition von Zielen beeinflusst die Definition eines Szenarios. Szenarien konkretisieren ein Ziel, indem sie einen beispielhaften Ablauf zur Erreichung eines Ziels beschreiben. Modellbasierte Anforderungen wie etwa Anforderungen modelliert in Klassendiagrammen initiieren die Definition von Szenarien, indem z.B. der konkrete Ablauf einer Methode erarbeitet wird. Szenarien wiederum dienen als wichtige Grundlage zur Validierung modellbasierter Anforderungen. Ziele beeinflussen den Ausschnitt der Realität, welche modellbasiert beschrieben werden soll. Die Inhalte eines ER-Diagramms hängen z.B. entscheidend davon ab, ob die Zielsetzung für die Repräsentation von Autos die Darstellung für den Kunden im Internet oder die Darstellung für die Buchhaltung ist.

Die zentrale Modellierung von Variabilität in Zielen, Szenarien und modellbasierten Anforderungen erfolgt durch die Relation von Varianten zu den entsprechenden Anforderungsartefakten. Dadurch wird die direkte Ableitung der Beziehungen der Anforderungsarten untereinander über die Beziehung der einzelnen Artefakte zur Variante ermöglicht. Abbildung 1 zeigt ein einfaches Beispiel für zentrale Variabilitätsmodellierung im Zusammenhang mit den 3 Arten von Anforderungen. Die Variante ‚Stau-meldung‘ wird mit dem Ziel Z₅, einem Teil des MSCs und der Anforderung R₁ in Beziehung gesetzt. Über die Beziehung zur Variante lassen sich die Beziehungen der Artefakte untereinander ableiten (graue Pfeile).

4 Benefits

Die Zuordnung unterschiedlicher Artefakte zu einer Variante und die Interrelationen zwischen den Anforderungsartefakten bietet folgende Vorteile:

- Verbesserte Kommunikation der Variabilität durch eigenständiges Modell.
- Variabilitätsmodell und Beziehung zu Anforderungsartefakten bieten unterschiedliche Einstiegspunkte für Stakeholder (z.B. Ziele, Szenarien für Management, ER-Diagramme für Entwickler).
- Ableitung von Beziehungen der Artefakte untereinander (Verbesserte Nachvollziehbarkeit)
- Konsistente Durchführung von Änderungen sowie Wiederverwendung durch Darstellung aller betroffenen Anforderungsartefakte.

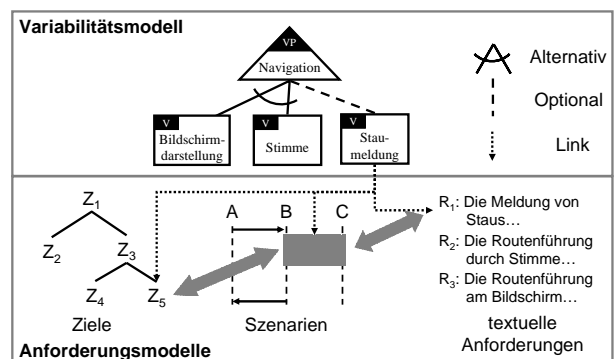


Abbildung 1: Beispiel Variabilitätsmodellierung

5 Ausblick

In unserer aktuellen Forschungsarbeit befassen wir uns im Moment mit den folgenden Themen:

- Change und Versionsmanagement durch das zentrale Variabilitätsmodell.
- Analyse von Anforderungsmodellen (bzgl. Verlinkung), z.B. UML 2.0.
- Werkzeugunterstützung für die orthogonale Variabilitätsmodellierung und Verbindung zu den 3 Arten von Anforderungen zusammen mit Evaluation des Ansatzes mit Industriepartnern

6 Referenzen

- [CN01] P. Clements, L. Northrop: Software Product Lines, Practices and Patterns; SEI Series in Software Engineering; Addison Wesley, 2001
- [BGL03] F. Bachmann, M. Goedicke, J. Leite, K. Pohl, B. Ramesh, A. Vilbig: Managing Variability in Product Family Development, Fifth International Workshop on Product Family Engineering, Siena, Italy, 2003
- [BLP04] S. Bühne, K. Lauenroth, K. Pohl: Why is it not Sufficient to Model Requirements Variability with Feature Models?; Workshop on Automotive Requirements Engineering in conjunction with RE'04, Nagoya, 2004
- [HP03] Günter Halmans, Klaus Pohl: Communicating the Variability of a Software Product Family to Customers, Software and Systems Modeling; Vol. 2, 15-36; Springer; Hamburg; 2003
- [Va02] Frank van der Linden: Software Product Families in Europe: The Esaps & Café Projects; IEEE Software, 19(4); 41-49, 2002