

Anforderungsorientierte Variabilitätsmodellierung für Software-Produktfamilien¹

Stan Bühne, Günter Halmans, Klaus Pohl

Software Systems Engineering,
Institut für Informatik und Wirtschaftsinformatik (ICB)
Universität Duisburg-Essen
Schützenbahn 70, 45117 Essen
Email: {buehne, halmans, pohl}@sse.uni-essen.de

Abstract:

Die Entwicklung von Software-Produktfamilien hat zum Ziel, durch Ausnutzung von Variabilität unterschiedliche Produkte auf der Basis einer gemeinsamen Plattform effizient und mit hoher Qualität zu entwickeln. Variabilität ist daher ein zentrales Konzept der Software-Produktfamilien Entwicklung. Für die Realisierung und Änderung von Produkten auf der Basis einer Software-Produktfamilie ist u. a. eine für alle Entwicklungsphasen (z.B. Requirements Engineering, Architekturdesign, Implementierung) durchgängige Repräsentation der Produktfamilien-Variabilität eine wesentliche Voraussetzung. Im Rahmen des Requirements Engineerings ist es notwendig, die Repräsentation der Variabilität mit unterschiedlichen Anforderungsmodellen wie beispielsweise Zielmodellen oder Szenarien zu verknüpfen und damit die verschiedenen Sichten der Anforderungsmodellierung einfließen zu lassen.

In diesem Beitrag beschreiben wir drei Arten von Anforderungsmodellen und deren Beziehungen untereinander. Im Weiteren erläutern wir, wie Produktfamilien-Variabilität in Bezug auf diese drei Arten von Anforderungsmodellen repräsentiert und die Produktdefinition durch die Beziehungen der Modelle untereinander vereinfacht werden kann.

1 Einleitung

Die Entwicklung von Software erfährt seit einigen Jahren die Schnellebigkeit der Märkte, in denen immer größere Softwareprojekte in immer kürzerer Zeit mit immer kleineren Budgets realisiert werden müssen. Um diesen Ansprüchen gerecht zu werden, wird vermehrt auf das Konzept der Software-Produktfamilien gesetzt [CN01].

Software-Produktfamilien haben für die Entwicklung neuer Softwareprodukte das Ziel, einen großen Teil der zugrunde liegenden Assets wieder zu verwenden, so dass individuelle Produkte einer Produktfamilie alle über einen gemeinsamen Teil (Common Assets) und über einen individuellen Teil (Variable Assets) verfügen, der einzelne Produkte

¹ Diese Arbeit wurde gefördert durch das BMBF Verbundprojekt CAFÉ „From Concept to Application in System Family Engineering“ (Förderkennzeichen 01 IS 002 C) und dem Europäischen ITEA Projekt FAMILIES „FACT-based Maturity through Institutionalisation Lessons-learned and Involved Exploration of System-family engineering“, Eureka ! 2023 Programme, ITEA Projekt ip02009.

voneinander unterscheidet. Assets einer Produktfamilie umfassen alle Teilprodukte der Entwicklung, z.B. Anforderungsspezifikationen, Modelle, Code-Komponenten oder Testfälle. Der Begriff Software-Produktfamilie und Produktfamilie wird im Folgenden synonym genutzt und meint in jedem Fall Software-Produktfamilien.

Software-Produktfamilien stützen sich auf zwei wesentliche Konzepte: zum einen auf die Aufteilung des Entwicklungsprozesses in Domain Engineering und Application Engineering und zum anderen auf die Produktfamilien-Variabilität.

Domain und Application Engineering: Im Domain Engineering (Konstruktionsprozess) werden die allgemeingültigen Produktfamilien-Assets und die Variationsmöglichkeiten definiert. Im Application Engineering (Produktionsprozess) werden unterschiedliche Produkte auf der Basis der Produktfamilien-Assets realisiert [Va02]. Ziel des Application Engineering ist es, individuelle Produkte durch Ausnutzung der Produktfamilien-Variabilität zu realisieren um damit einen hohen Wiederverwendungsgrad erreichen zu können.

Produktfamilien-Variabilität: Das Konzept der Variabilität ermöglicht es, Produkte auf Basis einer Produktfamilie zu erzeugen, die sich in bestimmten Ausprägungen voneinander unterscheiden. Produktfamilien-Variabilität wird unter anderem durch *Variationspunkte*, *Varianten* und *Abhängigkeiten zwischen Variationspunkten und Varianten* beschrieben (siehe auch [Bo01; BGL03]). Ein Variationspunkt beschreibt die Stelle innerhalb eines Assets, an der die Auswahl einer oder mehrerer Varianten möglich ist. Eine Variante beschreibt die konkreten (meist funktionalen) Möglichkeiten in Bezug auf einen oder mehrere Variationspunkte. Ein Variationspunkt besitzt eine oder mehrere Varianten. Varianten können von einem oder mehreren Variationspunkten abhängig sein. Für den Erfolg einer Software-Produktfamilie ist eine explizite Repräsentation der Produktfamilien-Variabilität essentiell. So können beispielsweise durch die explizite Repräsentation der Variabilität die Möglichkeiten der Produktfamilie dem Kunden kommuniziert und damit der Wiederverwendungsgrad gesteigert werden [HP03]. Des Weiteren ist eine explizite und über alle Entwicklungsphasen durchgängige Repräsentation Voraussetzung für die Nachvollziehbarkeit von Änderungen [BGL03].

In diesem Beitrag konzentrieren wir uns auf das Application Engineering. Basierend auf existierenden Ansätzen zur Modellierung von Variabilität zeigen wir, wie die Produktdefinition durch die Modellierung und explizite Repräsentation der Variabilität insbesondere im Zusammenhang mit unterschiedlichen Anforderungsmodellen unterstützt werden kann.

Kapitel 2 beschreibt einige Ansätze zur Modellierung der Produktfamilien-Variabilität. In Kapitel 3 gehen wir speziell auf die Modellierung der Variabilität für Anforderungen ein und erläutern einige sich daraus ergebende Frage- und Aufgabenstellungen. Kapitel 4 beschreibt unseren Ansatz zur Verknüpfung der Variabilitätsmodellierung mit drei Arten von Anforderungsmodellen (Zielmodelle, Szenarien, Lösungsorientierte Modelle) und erläutert die Vorteile dieses Ansatzes. Kapitel 5 fasst schließlich die Ergebnisse des Beitrages zusammen und gibt einen Ausblick.

2 Modellierung von Produktfamilien-Variabilität

In den folgenden Abschnitten beschreiben wir bisherige Arbeiten zur Repräsentation von Variabilität (Abschnitt 2.1) und die Berücksichtigung von Variabilität in unterschiedlichen Sichten des Entwicklungsprozesses (Abschnitt 2.2).

2.1 Repräsentation und Modellierung von Variabilität

Stefan Thiel und Andreas Hein beschreiben Variabilität auf der Ebene der Anforderungen durch Feature Modelle und Variabilität auf der Ebene der Architektur durch Variationspunkte in der Architektur [TH02]. Variationspunkte in der Architektur zeigen dabei die Architekturlösung der in den Feature Modellen angegebenen variablen Features. Der Hauptfokus dieser Arbeit liegt in der Betrachtung der Variabilität in der Architektur. Andere gängige Anforderungsmodelle, wie Zielmodelle oder Szenarien werden nicht betrachtet. Auch Fey et al. [FFB02] beschränken sich bei der Modellierung von Variabilität auf Featuremodelle. Clauß [CI01] beschreibt eine Erweiterung von UML-Modellen um Aspekte der Variabilität. Neben der Erweiterung der UML um neue Beziehungen und Stereotypen wird allerdings weder auf andere Anforderungsmodelle noch auf die Abhängigkeiten zwischen Assets aus Modellen anderer Entwicklungsphasen eingegangen. Felix Bachmann und Len Bass beschreiben in ihrer Arbeit die Repräsentation von Varianten innerhalb der Architektur und diskutieren unterschiedliche Realisierungsmöglichkeiten bezüglich dieser Varianten [BB01]. Jacobsen et al. führen Variationspunkte in Use Case-Modelle ein und beschreiben Realisierungsmöglichkeiten für Variabilität. Ihr Buch behandelt vor allem Variabilität in der Architektur [JGJ97].

Ansätze zur Modellierung von Variabilität aus der Sicht des Requirements Engineering werden durch eine Reihe von Arbeiten beschrieben. John und Muthig erläutern in ihrer Arbeit die Modellierung von Variabilität in Use Case Diagrammen und textuellen Use Case Beschreibungen [JM02]. Auch von der Maßen und Lichter behandeln die Repräsentation von Produktfamilien-Variabilität mit Use Cases [ML02]. Sie erweitern das UML-Metamodell zur Repräsentation von Variationspunkten. Schließlich fokussieren auch wir in einer früheren Arbeit die Repräsentation von Variabilität durch Use Cases [HP03], wobei der Fokus in dieser Arbeit speziell auf die Kommunikation der Variabilität zum Kunden und die damit notwendige explizite Repräsentation von Produktfamilien-Variabilität in Use Case Diagrammen liegt. Die aufgeführten Arbeiten bezüglich der Modellierung von Produktfamilien-Variabilität durch Use Cases betrachten allerdings keine alternativen Modellierungstechniken, wie Zielmodelle oder Lösungsorientierte Modelle. Des Weiteren werden in diesen Arbeiten z.B. Variationspunkte in die Anforderungsmodelle (hier Use Cases) integriert.

2.2 Verschiedene Sichten in der Produktfamilien Entwicklung

Bisherige Arbeiten zur Variabilität als zentrales Konzept von Produktfamilien haben deutlich gemacht, dass eine geeignete Repräsentation der Variabilität für das Auffinden, die Bearbeitung oder für die Kommunikation mit dem Kunden und damit für die Anforderungsmodellierung (hier Use Cases) integriert.

derungsgewinnung von essentieller Bedeutung ist [HP03]. Die Betrachtungen bezüglich der Repräsentation beziehen sich jeweils auf die unterschiedlichen Entwicklungsphasen wie z.B. das Requirements Engineering, die Architekturbeschreibung oder die Implementierung [BGL03]. Bachmann et al. schlagen in ihrer Arbeit vor, die spezielle Repräsentation von Produktfamilien-Variabilität aus den einzelnen Phasen herauszuziehen und in einer eigenen Variabilitätssicht zu modellieren (siehe Abbildung 1 und [BGL03]). Dabei werden durch entsprechende Beziehungen die Verknüpfungen zu den Assets der jeweiligen Phasen aufrechterhalten. Die Betrachtung der Variabilität in einer eigenen Sicht mit der Verknüpfung zu den jeweils betroffenen Assets aus Anforderungs- Architektur- und Implementierungssicht bietet dabei folgende Vorteile:

Die zum Teil sehr unterschiedlichen Modelle der einzelnen Entwicklungsphasen (z.B. Klassendiagramme für die Architektur oder Deployment-Diagramme für die Implementierung) müssen daher nicht für die Repräsentation von Produktfamilien-Variabilität erweitert werden. Damit können nicht standardisierte Ergänzungen der jeweiligen Notationen vermieden werden.

Durch die Repräsentation von Variabilität als eigene orthogonale Sicht kann eine durchgängige Nachvollziehbarkeit bezüglich der Variationspunkte, Varianten und ihren Abhängigkeiten untereinander über alle Entwicklungsphasen erreicht werden. Über einen Variationspunkt ist so beispielsweise erkennbar, welche Anforderungsvarianten mit welchen Architektur-Elementen bzgl. des Variationspunktes in Beziehung stehen und ob für diese Anforderung ggf. auf der Ebene der Architektur zusätzliche Variabilität existiert (siehe Abbildung 1).

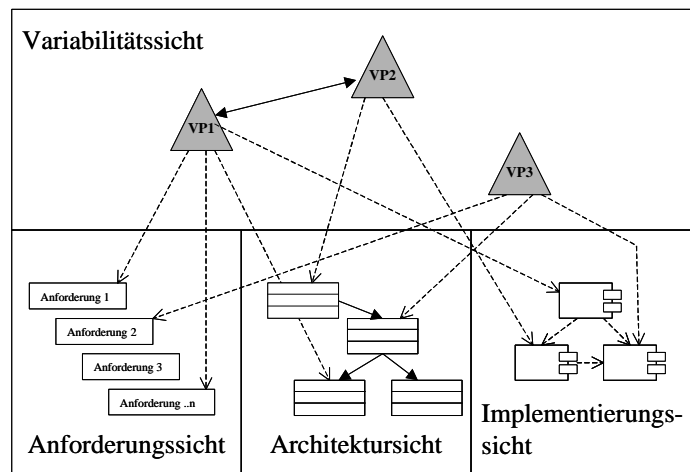


Abbildung 1: Variabilitäts-Sicht als orthogonale Sicht zu Anforderungen, Architektur und Implementierung

In dem von Bachmann et al. vorgeschlagenen Metamodell zur Repräsentation von Variabilität [BGL03] wird in der Variabilitätssicht eine Variante einem oder mehreren Assets zugeordnet. Anders ausgedrückt: durch die Zuordnung von Assets zu einem Variationspunkt werden Assets zu Varianten des Variationspunktes. Mit Hilfe dieser Beziehung kann für einen Variationspunkt nachvollzogen werden, welche Assets – unabhängig aus

welcher Entwicklungsphase es stammt – relevant sind. Umgekehrt kann für jedes Asset ermittelt werden, mit welchen Variationspunkten es in Beziehung steht (d.h. warum bzw. wo es variiert). Ein Variationspunkt hat darüber hinaus einen Grund (Rationale), welcher beschreibt, warum dieser Variationspunkt existiert (z.B. existiert ein Variationspunkt Sprache aus dem Grund, dass ein System in verschiedenen Ländern angeboten werden soll).

Becker et al. führen in ihrer Arbeit das Konzept der *Generischen Software Artefakte* (generic software artefacts - GSA) ein [Be01]. Sie führen an, dass die GSA's Produkte aus allen Bereichen der Entwicklung wie etwa Architektur-Sichten, Komponenten-Code oder Anforderungsmodelle umfassen können. Variationspunkte sind in ihrem Variabilitätsmodell Teile der GSA's. Die Einführung der GSA's ähnelt damit sehr der Einführung von Assets in den Arbeiten von Bachmann et al. [BGL03].

In [GB02] beschreiben Geyer und Becker ein Variabilitätsmodell, welches orthogonal zu den unterschiedlichen Entwicklungsphasen des Application Engineerings steht. Sie definieren Variabilität als die Möglichkeit, ein System zu verändern bzw. verschiedene Einstellungen vornehmen zu können (Customization). Variabilitäten repräsentieren ihrer Definition nach variable Aspekte, die mit einem oder mehreren Variationspunkten in Beziehung stehen. Im Gegensatz zum Modell von Bachmann et al., in der die Assoziation zwischen der Variante und den Assets gebildet wird (Abbildung 2), stellen Geyer und Becker Variationspunkte und Assets miteinander in Beziehung. Bei Geyer und Becker wird die Verbindung zu den Assets aus den verschiedenen Entwicklungsphasen (Anforderungen, Architektur, Implementierung) über Variationspunkte hergestellt. Der Hauptfokus ihrer Arbeit liegt hierbei in der Beschreibung des Einflusses der Variabilität auf den Produktionsprozess (Application Engineering) und sich daraus ergebende Anforderungen an eine Konfigurationstechnik zur Ableitung konkreter Produkte [GB02]. Demgegenüber beschreibt die Arbeit von Bachmann et al. [BGL03] die Entwicklung der Variabilität von der Anforderung bis hin zur Implementierung und macht deutlich, dass z.B. auf Grund von Designentscheidungen weitere Variationspunkte hinzukommen können.

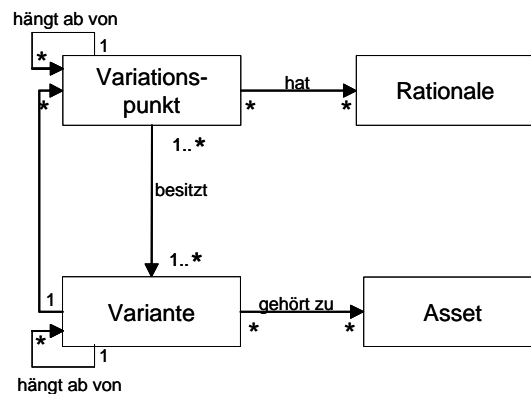


Abbildung 2: Auszug aus dem Variabilitäts-Metamodell von [BGL03]

3 Modellierung der Variabilität von Anforderungen

Die in [BGL03] vorgeschlagene orthogonale Variabilitäts-Sicht sowie das zugehörige Metamodell stellen in abstrakter Form die Beziehung zwischen der Variabilität und den Assets aus den verschiedenen Entwicklungsphasen dar. Diese Assets können in den einzelnen Entwicklungsphasen vielfältig sein (siehe Abbildung 3). So werden im Rahmen der Architektursicht bspw. Klassen- oder Kollaborationsdiagramme eingesetzt.

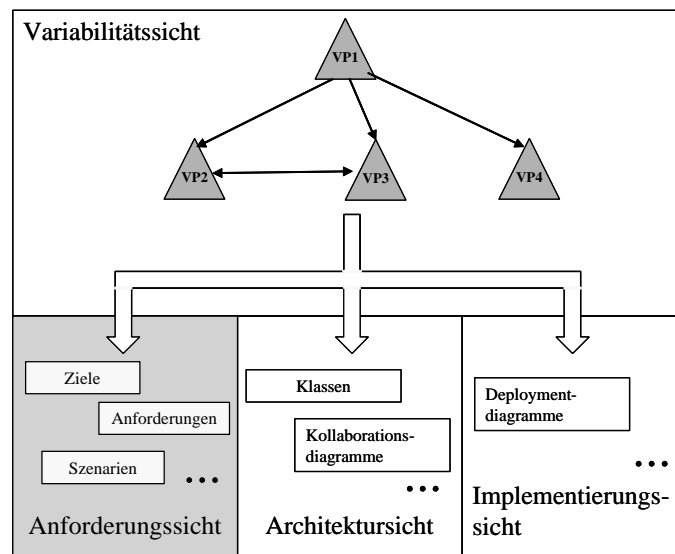


Abbildung 3: Modellierung der Variabilität in Bezug auf die Anforderungssicht

Motiviert durch die besondere Bedeutung der frühzeitigen Berücksichtigung von Produktfamilien-Variabilität bereits in der Produktdefinitionsphase [CN01; HP03], konzentrieren wir uns im Folgenden auf die Modellierung der Variabilität in der Anforderungssicht. Die unterschiedlichen Elemente in der Anforderungssicht nennen wir *Anforderungs-Assets*. Wir setzen dabei folgende Anforderungsmodelle ein: *Zielorientierte Modelle*, *Szenarien* und *Lösungsorientierte Modelle*.

Zielorientierte Modelle dokumentieren die mit dem betrachteten System verbundene Vision und verfeinern diese Vision durch weitere Unterziele. Zielorientierte Modelle sind Grundlage für ein zielorientiertes Vorgehen und unterstützen den Einigungsprozess unter den am Produkt interessierten Personen auf abstrakter Ebene. Zielmodelle sind in der Regel übersichtlich und beschreiben den Mehrwert eines Systems. Zielorientierte Modelle repräsentieren oftmals die Managementsicht in Bezug auf das betrachtete System.

Szenarien beschreiben konkrete Anwendungsfälle. Wir verstehen Szenarien hier als eine verallgemeinerte Form von Use Cases. Sie stellen konkrete Benutzer/System-Interaktionen dar, die zur Erfüllung bzw. Nichterfüllung eines Zieles führen. Szenarien konkretisieren somit Ziele und führen sehr oft auch zu neuen (Sub-)Zielen. Szenarien repräsentieren die Sicht des Systemnutzers.

Lösungsorientierte Modelle, z.B. ER-Diagramme, definieren die Anforderungen aus Sicht der Softwareentwicklung. Sie beinhalten oft die Beschreibung einer prinzipiellen Lösung wie bspw. grobe Datenstrukturen oder das Verhalten von Objekten. Die konkrete Lösung selbst, also etwa das realisierte System, wird hierunter nicht verstanden. Im Folgenden betrachten wir als Beispiel eines lösungsorientierten Modells strukturierte, textuelle Beschreibungen von Anforderungen mit vorgegebenen Attributen.

Die Anwendung der einzelnen Modellierungstechniken ist in Theorie und Praxis vielfach erprobt. Für die Modellierung der Interaktionen eines Systems mit dem Anwender sowie für die Kommunikation der Funktionalität zum Anwender werden Szenarien / Use Cases sehr erfolgreich eingesetzt [Ja92; Ca95; We98]. Lösungsorientierte Modelle wie ER-Diagramme oder auch verschiedene Diagramme der UML finden in vielen Projekten der Softwareentwicklung Verwendung. Schließlich dienen Zielmodelle vor allem der strukturierten Dokumentation der Systemziele und deren Dekomposition zu verfeinerten Subzielen. Zielmodelle gewinnen immer mehr an Bedeutung, wie z.B. van Lamsweerde deutlich macht: „*Requirements engineering research has increasingly recognized the leading role played by goals in the RE process*“ [LA01], S249.

Für das Requirements Engineering und speziell für die Produktdefinitionsphase im Rahmen des Application Engineering ergeben sich bezüglich der Produktfamilien-Variabilität und der einzelnen Anforderungsmodelle u.a. folgende Fragen:

Angenommen, die Produktfamilie erfüllt die optionalen Ziele A und B. Welche konkreten Abläufe erfüllen diese Ziele? Z.B. unterstützt ein Hausüberwachungssystem die zwei optionalen Ziele *internes Löschen* (d.h. bei einem Brand sollen eigenständige Mechanismen zum Löschen des Brandes aktiviert werden, z.B. durch Sprinkleranlagen) und *externes Löschen* (d.h. bei einem Brand sollen sofort externe Löschkraften aktiviert werden, z.B. durch eine Direktleitung zur Feuerwehr). Für den Anwender (Kunden) stellt sich hier die Frage, welche konkreten Abläufe zur Erfüllung der jeweiligen Ziele führen.

Gibt es bezogen auf einen konkreten Ablauf Varianten, die ebenfalls zur Erfüllung der Kundenanforderungen führen oder neue Möglichkeiten aufdecken?

Lässt sich die Menge der Möglichkeiten, die ausgewählt werden können, sinnvoll eingrenzen? Eine Produktfamilie kann leicht mehrere hundert Variationspunkte in Anforderungs-Assets mit entsprechenden Varianten enthalten, so dass eine Komplexitätsreduktion im Rahmen der Produktdefinitionsphase von essentieller Bedeutung ist.

Aus der Betrachtung dieser und ähnlicher Fragen sowie der Einführung einer orthogonalen Variabilitätssicht (siehe Abschnitt 2.2) ergeben sich in Bezug auf die Anforderungssicht folgende Aufgabenstellungen:

Die Modellierung der Variabilität für die Anforderungs-Assets soll wie in [BGL03] vorgeschlagen in einer eigenen Variabilitätssicht erfolgen. Damit wird die durchgängige Nachvollziehbarkeit über die verschiedenen Entwicklungsphasen (Requirements Engineering, Architekturdesign und Implementierung) erreicht.

Die Beziehungen zu den Anforderungs-Assets aus der Anforderungssicht müssen modelliert werden. Diese Anforderungs-Assets können beispielsweise Szenarien oder textuelle Beschreibungen von Anforderungen umfassen.

Bisher gibt es keine Arbeiten über die Verbindung einer wie in [BGL03] beschriebenen, orthogonalen Variabilitäts-Sicht zu den verschiedenartigen Assets aus der Anforderungssicht und welche Auswirkungen sich daraus ergeben. Bisherige Arbeiten fokussieren meist nur Teilaspekte (wie etwa der Repräsentation von Variabilität in Use Cases [JGJ97; ML02; JM02; HP03], siehe Abschnitt 2.1) oder beschreiben Variabilitätsmodelle auf abstraktem Level ohne diese weiter zu instanzieren (Abschnitt 2.2).

4 Ein Variabilitätsmodell für die drei Anforderungsmodelle

In diesem Kapitel beschreiben wir ein Variabilitätsmodell für die Anforderungssicht. Abschnitt 4.1 erläutert die Beziehungen zwischen den in Kapitel 3 vorgestellten Anforderungsmodellen. In Abschnitt 4.2 beschreiben wir unseren Ansatz einer auf dem in Abschnitt 2.1 vorgestellten Metamodell basierenden Repräsentation der Produktfamilien-Variabilität, die wir mit den drei Arten von Anforderungsmodellen verknüpfen. Abschnitt 4.3 führt die sich aus dem Ansatz ergebenden Vorteile insbesondere für die Produktdefinition im Rahmen des Application Engineerings auf.

4.1 Beziehungen zwischen den drei Arten von Anforderungsmodellen

Der Einsatz der drei Arten von Anforderungsmodellen (siehe Kapitel 3) im Requirements Engineering ist nicht nur in Bezug auf jedes einzelne Anforderungsmodell sinnvoll. Ein wesentlicher Vorteil der Anwendung der Anforderungsmodelle liegt in der Berücksichtigung ihrer gegenseitigen Beziehungen. Abbildung 4 stellt die drei Modelle für Anforderungen sowie die Beziehungen untereinander dar. Diese Beziehungen lassen sich wie folgt näher beschreiben:

Beziehung zwischen Szenarien und Zielorientierten Modellen: Die Definition von Zielen initiiert und beeinflusst die Definition eines Szenarios. Szenarien konkretisieren ein Ziel, indem sie einen beispielhaften Ablauf beschreiben, wie das Ziel erreicht werden kann. Szenarien führen durch ihren konkreten Charakter oftmals zu verfeinerten Zielen.

Beziehung zwischen Szenarien und Lösungsorientierten Modellen: Lösungsorientierte Modelle wie Klassendiagramme initiieren die Definition von Szenarien, indem zum Beispiel der konkrete Ablauf einer Methode erarbeitet wird. Szenarien wiederum dienen als wichtige Grundlage zur Validierung lösungsorientierter Modelle. Die in den Szenarien beschriebenen Abläufe müssen von den Lösungsmodellen erfüllt werden können.

Beziehung zwischen Zielen und Lösungsorientierten Modellen: Ziele beeinflussen den Ausschnitt der Realität, welche lösungsorientiert modelliert 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. Lösungsorientierte Modelle wiederum führen zu Verfeinerungen und zur Neuentdeckung von Zielen, beispielsweise wenn sich herausstellt, dass mit der Definition eines ER-Diagrammes für ein Cont-

rolling-System bisher nicht bedachte Auswertungen ermöglicht und neue Controlling-Ziele verfolgt werden können.

Durch die existierenden Beziehungen und die gegenseitige Beeinflussung der Modelle bzw. der damit verbundenen Sichten wird die Anforderungsdokumentation vollständiger und besser nachvollziehbar. So lässt sich mit der Verknüpfung von Szenarien zu einem Ziel begründen, wieso ein bestimmtes Szenario und die damit verbundenen Anforderungen in dem System umgesetzt werden (nämlich zur Erfüllung des intendierten Zieles).

Durch die Nutzung der hier aufgeführten Beziehungen zwischen den Anforderungsmodellen kann die Produktdefinitionsphase im Zusammenhang mit einem Variabilitätsmodell für Anforderungen unterstützt werden (siehe Abschnitt 4.2 und 4.3).

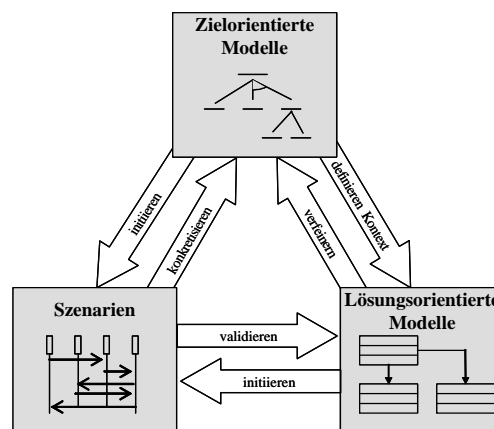


Abbildung 4: Drei Modelle von Anforderungen

4.2 Modellierung der Variabilität für die drei Anforderungsmodelle

In diesem Abschnitt beschreiben wir unseren Ansatz, die Repräsentation der Variabilität basierend auf dem Metamodell zur Modellierung von Variabilität ([BGL03]) auf die beschriebenen Anforderungsmodelle anzuwenden und die in Abschnitt 4.1 genannten Verknüpfungen der Anforderungsmodelle untereinander im Hinblick auf die Produktdefinition zu nutzen. Im Einzelnen erläutern wir die Repräsentation der Produktfamilien-Variabilität aus der Sicht von (1) Zielorientierten Modellen, (2) Szenarien und (3) Lösungsorientierten Modellen (strukturierten textuellen Beschreibungen von Anforderungen). Die Beispiele zur Erläuterung der Variabilität bezüglich der einzelnen Anforderungsmodelle beziehen sich auf die Domäne *Intelligentes Gebäude*. Im intelligenten Gebäude wird durch Vernetzung der Gebäudetechnik und die Einbettung von Software die Funktionalität z.B. bezüglich der Überwachung oder Steuerung von technischen Geräten erheblich erweitert.

(1) Variabilität aus der Sicht von Zielorientierten Modellen

Variabilität aus der Sicht von Zielorientierten Modellen bedeutet, dass für die Definition eines Produktes die zu erfüllenden Ziele aus einem bestimmten Grund (z.B. in Bezug auf ein übergeordnetes Ziel) wählbar sind. Beispielsweise kann ein Ziel A ei-

ne Alternative zu einem Ziel B sein oder aber Ziel A und Ziel B sind Teilziele von Ziel XY (siehe Abbildung 5), wobei während der Produktdefinition entschieden werden kann, ob das Produkt das Teilziel B erfüllen soll, oder nicht. Ein Ziel kann dabei durchaus zu mehreren Variationspunkten eine Verbindung haben und dadurch Variante von mehreren Variationspunkten sein (siehe Variante B in Abbildung 5).

In einem intelligenten Gebäude kann zum Beispiel eine Meldung im Störfall für bestimmte Geräte der Haustechnik ein Ziel beschreiben, welches der Auftraggeber mit dem Produkt „intelligentes Gebäude“ verknüpft. Je nach Bedarf werden mit dem Ziel der Störmeldung unterschiedliche und optionale Unterziele verknüpft sein. Zum Beispiel kann ein Unterziel darin liegen, die Störmeldung jederzeit über ein entsprechendes Medium abfragen zu können (passive Störmeldung) oder zum anderen aktiv an eine bestimmte Adresse (z.B. per SMS an ein Mobiltelefon) zu schicken.

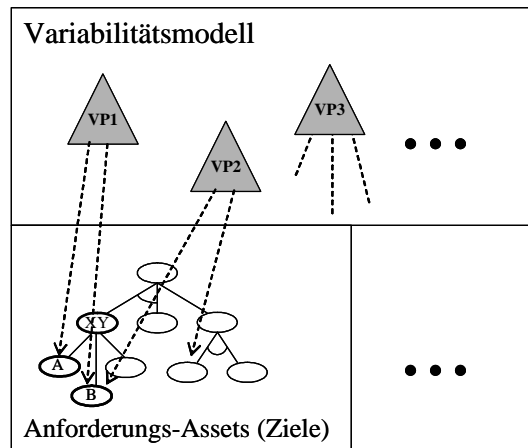


Abbildung 5: Variabilitätsmodell bezüglich Ziele

Die Verknüpfung der beiden optionalen Ziele (passive und aktive Störmeldung) ermöglicht im Rahmen der Produktdefinition und der Diskussion mit dem potentiellen Kunden eine direkte Konkretisierung der Ziele. Beispielsweise wird das Ziel *passive Störmeldung* mit einem Szenario verknüpft sein, welches beschreibt, wie man über das Internet die Statusanzeige des Gebäudes aktivieren und somit evtl. Störungen anzeigen lassen kann. Demgegenüber wird das Ziel einer aktiven Störmeldung durch ein Szenario erfüllt, in dem der Hausherr mittels einer SMS aktiv über die Störung in seinem Gebäude informiert wird.

Ist ein Ziel mit mehreren Szenarien verknüpft, so bedeutet dies, dass es verschiedene Abläufe zur Erfüllung eines Ziel gibt. In diesem Falle wird ein Variationspunkt im Variabilitätsmodell bezüglich der beiden Szenarien eingeführt, so dass die hier beschriebene Variabilität explizit wird.

Durch die Verknüpfung von Zielen zu Lösungsorientierten Modellen (in diesem Falle strukturierten, textuellen Anforderungen) können die wählbaren Ziele weiter konkretisiert werden. Es wird verdeutlicht, wie eine Lösung aussehen kann, die diese Ziele erfüllen wird. Anforderungen bezüglich des Ziels aktive Störungsmeldung (und

des entsprechenden Szenarios) wären beispielsweise: *Es muss möglich sein, die Nummer eines Mobiltelefonanschlusses einzugeben, und das System muss in der Lage sein, einen Text in SMS-Format an einen definierten Mobiltelefonanschluss zu senden.*

Die Repräsentation der Variabilität in Bezug auf Ziele ermöglicht eine abstrakte Betrachtung der Produktfamilienmöglichkeiten. Mit der Auswahl eines Zieles wie etwa der passiven Störmeldung lässt sich das Auswahlverfahren bezüglich potentieller Variationspunkte erheblich vereinfachen. In diesem Beispiel ist es nicht mehr notwendig, sich über Varianten zur aktiven Störmeldung (SMS, Meldung an die Polizei bzw. Feuerwehr etc.) weitere Gedanken zu machen. Die abstrakte Sicht der Ziele ist z.B. für das Management interessant. Auf der Basis von Entscheidungen bezüglich der Zielvarianten kann die Grobstruktur des zu entwickelnden Produktes bereits festgelegt werden, ohne sich über die Auswahl von Varianten auf sehr detaillierter Ebene im Klaren sein zu müssen.

(2) Variabilität aus der Sicht von Szenarien

In einer Produktfamilie können Szenarien als Varianten definiert werden. Ein Szenario A beschreibt beispielsweise einen anderen Ablauf als ein Szenario B, jedoch liefern beide Szenarien zum Beispiel das gleiche Ergebnis. In diesem Falle können die Szenarien einem Variationspunkt zugeordnet und damit als Variante geführt werden (ähnlich wie die Ziele A und B in Abbildung 5 dem Variationspunkt VP1 zugeordnet wurden).

Ein Szenario A zur Überwachung des Beleuchtungssystems im intelligenten Gebäude beschreibt, wie im Gebäude an einem lokalen Steuerungssystem die einzelnen Räume nach dem Status ihrer Beleuchtungssysteme abgefragt werden. Ein anderes Szenario B wiederum stellt die Überwachung über einen Fernzugang (z.B. über das Internet) dar. In diesem Szenario wird z.B. eine Authentifizierung integriert sein, so dass Szenario B sich von Szenario A unterscheidet.

Die Wertschöpfung der optionalen Szenarien kann über die Verknüpfung zu den Zielen, welche die Produktfamilie erfüllen kann, abgeleitet werden. Im intelligenten Gebäude gibt es beispielsweise die beiden Szenarien zur Störmeldung. Die optionalen Szenarien A und B bezüglich der Überwachung des Beleuchtungssystems dienen dem Ziel einer lokalen Steuerungs- und Überwachungseinheit (Szenario A) sowie dem Ziel, den Status des Gebäudes (bzw. seiner Technik) von außen bzw. von beliebigen Orten abrufen zu können (Szenario B).

Die Verknüpfung der optionalen Szenarien zu Lösungsorientierten Modellen (in diesem Falle strukturierten, textuellen Anforderungen) ermöglicht einen Ausblick darauf, wie diese Szenarien konkret umgesetzt werden können. Beispielsweise wird die Authentifizierung in Szenario B durch die Anforderungen *der User muss sich am System mit einem Login und einem Passwort anmelden können und nach dreimaligem Fehlversuch ist eine erneute Anmeldung nicht mehr möglich* weiter detailliert.

Die Repräsentation der Variabilität in Bezug auf Szenarien beschreibt insbesondere die Variationsmöglichkeiten aus der Sicht des Benutzers, da hier vor allem die Variabilität im Rahmen der Interaktion mit dem System beschrieben wird. Die Betrachtung der Variabilität bezüglich Szenarien ermöglicht einen konkreten Einblick in die unterschiedlichen, von der Produktfamilie ermöglichten Abläufe. Diese Sicht wurde

bereits in verschiedenen Arbeiten [JM02; ML02; BHP03; HP03] untersucht, wobei hierbei in der Regel eine Integration von Variationspunkten in die genutzten Anforderungsmodelle (z.B. Use Cases) vorgenommen wurde.

(3) Variabilität aus der Sicht von Lösungsorientierten Modellen

Die Repräsentation von Variabilität aus der Sicht von Lösungsorientierten Modellen ermöglicht eine weitere, detaillierte Beschreibung möglicher Varianten. Eine Lösungsvariante A beschreibt einen anderen Ansatz als die Lösungsvariante B, beide können aber zu demselben Ergebnis führen.

Für die Aktivierung eines Beleuchtungssystems im intelligenten Gebäude gibt es die beiden folgenden Anforderungen: *das Beleuchtungssystem muss mit der Hand über einen Schalter ein- und ausschaltbar sein*. Eine weitere optionale Anforderung dazu ist die folgende: *das Beleuchtungssystem muss per Spracheingabe ein- und ausschaltbar sein*. Ein System zur Unterstützung des intelligenten Gebäudes kann eines oder beide Anforderungen erfüllen.

Wie bei den Szenarien beschreibt die Verknüpfung zu Zielen den Kontext der Anforderung und die durch die Umsetzung der Anforderung erzielte Wertschöpfung. Ein Ziel für das Aktivieren des Leuchtsystems mit der Hand liegt in der Abbildung traditioneller Funktionen, so dass sich auch Personen, die sich sonst in einem „normalen“ Gebäude aufhalten, zurechtfinden. Das Ziel der Spracheingabe ist zum Beispiel die Unterstützung behindertengerechten Wohnens. Verknüpfte Szenarien verdeutlichen ebenfalls den Kontext der unterschiedlichen Anforderungsvarianten und beschreiben, in welchem Ablauf die jeweilige Variante genutzt wird.

Mit der Modellierung von Variabilität in Bezug auf Lösungsorientierte Modelle werden Alternativen und Optionen hinsichtlich konkreter Lösungsvorschläge (damit ist nicht die Lösung im Sinne der technischen Umsetzung bzw. der Implementierung gemeint) verdeutlicht. Die explizite Darstellung alternativer und optionaler Umsetzungsmöglichkeiten ist z.B. für die Entwicklung von Interesse.

Die Beziehungen zwischen Anforderungs-Assets und Variationspunkten können verschiedene Eigenschaften haben. Zum Beispiel kann eine Beziehung bedeuten, dass aus einer Menge von Anforderungs-Assets in Bezug auf einen Variationspunkt eines oder mehrere Assets ausgewählt werden müssen (mandatory), andere aber optional ausgewählt werden können. Die Art dieser Beziehungen zu einem Variationspunkt wird in anderen Arbeiten (u.a. [HP03; BHP03]) ausführlich diskutiert. Auf die Beziehungen zwischen Variationspunkten und den beschriebenen Anforderungs-Assets sind die Eigenschaften prinzipiell übertragbar.

4.3 Vorteile der Modellierung von Produktfamilien-Variabilität in Bezug auf die drei Anforderungsmodelle

Der in Abschnitt 4.2 beschriebene Ansatz zur Anwendung der Variabilitätsmodellierung auf Zielorientierte und Lösungsorientierte Anforderungsmodelle sowie Szenarien bietet für die Produktdefinition folgende Vorteile:

Variationspunkte in Zielen stellen oftmals eine abstrakte Repräsentation der Variabilität dar. Es ist dadurch möglich, bereits auf hohem Abstraktionsniveau eine Auswahl für das zu definierende Produkt zu treffen. Damit können Variationspunkte von Varianten, die nicht mit dem ausgewählten Ziel verbunden sind, im weiteren Entscheidungsprozess herausfallen.

Durch die Verknüpfung von Zielen mit Szenarien und Lösungsorientierten Modellen können in der Produktdefinition optionale Ziele direkt konkretisiert werden.

Durch die Verknüpfung von Szenarien mit Zielen kann die Wertschöpfung von optionalen Szenarien beschrieben werden. Darüber hinaus wird der Kontext des Szenarios (in welchem Zusammenhang soll das Szenario eingesetzt werden) verdeutlicht.

Durch die orthogonale Repräsentation der Variabilität in einer eigenen Sicht können beispielsweise Ziele (bzw. auch andere Anforderungs-Assets) mehreren Variationspunkten zugeordnet werden, ohne dass das eigentliche Zielmodell (bzw. Szenario oder Lösungsorientierte Modell) verändert wird.

Angesichts der Tatsache, dass eine Produktfamilie alleine bezüglich der Anforderungs-Assets Hunderte von Varianten beinhalten kann, stellt die Einschränkung von möglichen Varianten durch eine Auswahl auf höherem Abstraktionsniveau (einem Ziel) eine erhebliche Vereinfachung innerhalb der Produktdefinitionsphase dar. Die Diskussion mit dem Kunden über Zielvarianten, die sein zukünftiges Produkt erfüllen sollen, wird durch die Konkretisierung mit verknüpften Szenarien unterstützt.

5 Zusammenfassung und Ausblick

Die Entwicklung von Software-Produktfamilien hat zum Ziel, durch Ausnutzung von Variabilität unterschiedliche Produkte auf der Basis einer gemeinsamen Plattform effizient und mit hoher Qualität zu entwickeln. Variabilität ist daher ein zentrales Konzept der Software-Produktfamilien Entwicklung. Für die einzelnen Entwicklungsphasen ist eine durchgängige und explizite Repräsentation der Produktfamilien-Variabilität essentiell. Nur so können z.B. die sich aus der Variabilität ergebenden Möglichkeiten der Produktfamilie kommuniziert oder im Vergleich zur Variabilität in Anforderungen zusätzliche Variabilität im Bereich der Architektur dokumentiert werden. Eine Repräsentation der Variabilität in der Entwicklungsphase „Requirements Engineering“ erfordert dabei die Berücksichtigung unterschiedlicher Anforderungsmodelle.

In diesem Beitrag konzentrieren wir uns auf das Application Engineering und eine mögliche Unterstützung für die Produktdefinitionsphase. Wir beschreiben basierend auf dem Ansatz von [BGL03] zur Modellierung von Variabilität für unterschiedliche Entwicklungsphasen die Repräsentation der Produktfamilien-Variabilität für die Anforderungssicht. Weiterhin gehen wir auf drei wesentliche Arten der Anforderungsmodellierung und ihre Beziehungen untereinander ein: Zielmodelle, Szenarien und Lösungsorientierte Modelle. In unserem Ansatz beschreiben wir eine durchgängige und explizite Repräsentation der Produktfamilien-Variabilität für die drei Anforderungsmodelle. Diese orthogonale Repräsentation der Variabilität für die Anforderungsmodelle führt in Verbindung

mit den Beziehungen der Anforderungsmodelle untereinander zu den folgenden Vorteilen für die Produktdefinition:

Ziele ermöglichen eine abstraktere Darstellung von Varianten und lassen frühzeitige Einschränkungen durch die Auswahl von Zielen zu.

Szenarien beschreiben konkrete Abläufe, zum Beispiel bezüglich eines Zieles, und beschreiben die Funktionalität der unterschiedlichen Varianten

Lösungsorientierte Modelle detaillieren Ziele sowie Szenarien und beschreiben mögliche Umsetzungen bezüglich der betrachteten Varianten

Das zu den Anforderungsmodellen orthogonale Variabilitätsmodell vermeidet Erweiterungen in den Anforderungsmodellen und bildet die Voraussetzung für eine über die verschiedenen Entwicklungsphasen hinweg durchgängige Repräsentation

Gegenstand unserer aktuellen Arbeiten ist die Betrachtung von Beziehungen und Hierarchien zwischen den Variationspunkten des Variabilitätsmodells. Durch die Einführung abstrakter Variationspunkte können z.B. zwei Variationspunkte VP1 und VP2 miteinander in Beziehung gesetzt werden. Dadurch lassen sich Rückschlüsse auf die mit den Variationspunkten VP1 und VP2 verknüpften Varianten ziehen und auswerten.

Die in diesem Beitrag diskutierten Verknüpfungen zwischen unterschiedlichen Anforderungsmodellen sowie ihren Beziehungen zu einer orthogonalen Variabilitätssicht führen zu einer hohen Komplexität z.B. bezüglich der konsistenten Einarbeitung von Änderungen. Eine entsprechende Unterstützung durch entsprechende Werkzeuge ist somit unabdingbar. Die Visualisierung der in diesem Beitrag beschriebenen Variabilität für Anforderungen inklusiver ihrer Verknüpfungen sowie eine Unterstützung durch geeignete Werkzeuge ist Gegenstand unserer zukünftigen Forschungsarbeiten.

Referenzen

- [BB01] Felix Bachmann, Len Bass: *Managing Variability in Software Architecture*; ACM Press, NY, USA, 2001
- [Be01] M. Becker, L. Geyer, A. Gilbert, K. Becker: *Comprehensive Variability Modelling to Facilitate Efficient Variability Treatment*; Proceedings of the 4th Workshop on Product Family Engineering (PFE-04), Bilbao, Spain, 2001
- [BGL03] Felix Bachmann, Michael Goedicke, Julio Leite, Klaus Pohl, Balasubramaniam Ramesh, Alexander Vilbig: *Managing Variability in Product Family Development*, Fifth International Workshop on Product Family Engineering (PFE5), Siena, Italy, 4-6 November 2003
- [BHP03] Stan Bühne, Günter Halmans, Klaus Pohl: *Modelling Dependencies between Variation Points in Use Case Diagrams*, Ninth International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ 03), Velden, Österreich, Juni 2003
- [Bo01] Jan Bosch, Gert Florijn, Danny Greefhorst, Juha Kuusela, Henk Obbink, Klaus Pohl: *Variability Issues in Software Product Lines*; Fourth International Workshop on Product Family Engineering (PFE-4), Bilbao, Spain, 2001
- [Ca95] J. Carroll: *The Scenario Perspective on System Development, Scenario-Based Design: Envisioning Work and Technology in System Development*; John Wiley & Sons, 1995
- [Cl01] Matthias Clauß: *Generic Modeling using UML Extensions for Variability*; Workshop on Domain Specific Visual Languages: An OOPSLA 2001 Workshop, Tampa Bay, Florida

- [CN01] Paul Clements, Linda Northrop: Software Product Lines, Practices and Patterns; SEI Series in Software Engineering; Addison Wesley, 2001
- [FFB02] Dániel Fey, Róbert Fajta, and Anreás Boros: Feature Modeling: A Meta-model to Enhance Usability and Usefulness; Software Product Lines; Proceedings of the Second International Conference, SPLC 2; Springer; G. J. Chastek (Ed.), San Diego, USA, August 2002
- [GB02] Lars Geyer, Martin Becker: On the Influence of Variabilities on the Application-Engineering Process of a Product Family; Software Product Lines; Proceedings of the Second International Conference, SPLC 2; Springer; G. J. Chastek (Ed.), San Diego, USA, August 2002
- [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; März 2003
- [Ja92] Ivar Jacobson: Object-Oriented Software Engineering: A Use Case Driven Approach; Addison Wesley, 1992
- [JGJ97] Ivar Jacobson, Martin Griss, Patrik Jonsson: Software Reuse, Architecture, Process and Organization for Business Success; Addison Wesley, 1997
- [JM02] Isabel John, Dirk Muthig: Tailoring Use Cases for Product Line Modeling; Proceedings of the International Workshop on Requirements Engineering for Product Lines 2002 (REPL'02), Technical Report: ALR-2002-033, AVAYA labs, 2002
- [LA01] Axel van Lamsweerde: Goal-Oriented Requirements Engineering: A Guided Tour; 5th IEEE International Symposium on Requirements Engineering, Proceedings RE'01, Toronto 2001
- [ML02] Thomas von der Maßen, Horst Lichter: Modeling Variability by UML Use Case Diagrams; Proceedings of the International Workshop on Requirements Engineering for Product Lines 2002 (REPL'02), Technical Report: ALR-2002-033, AVAYA labs, 2002
- [TH02] Stefan Thiel, Andreas Hein: Modeling and Using Product Line Variability in Automotive Systems, IEEE Software, July/August 2002, p 66ff.
- [We98] Klaus Weidenhaupt, Klaus Pohl, Matthias Jarke, Peter Haumer: Scenario Usage in System Development: A Report on Current Practice IEEE Software, 15(2): 34-45, March/April 1998
- [Va02] Frank van der Linden: Software Product Families in Europe: The Esaps & Café Projects; IEEE Software, 19(4): 41-49, July/August 2002