Paradigmen zur Variabilitätsbeschreibung von Vorgehensmodellen

Christian Bartelt, Edward Fischer, Thomas Ternité

Institut für Informatik
Technische Universität Clausthal
{christian.bartelt | edward.fischer | thomas.ternite}@tu-clausthal.de

1 Einleitung

Viele Unternehmen und Organisationen haben erkannt, dass systematisch geplante und gelebte Prozesse die Effizienz und Qualität signifikant steigern. Allerdings sind der Entwurf und die Einführung von umfassenden, organisationsspezifischen Vorgehensmodellen oft sehr kosten- und zeitaufwändig.

Um die Einführung von Vorgehensmodellen im IT-Umfeld zu unterstützen, wurden in den letzten Jahren eine Reihe von Prozess- und Vorgehensstandards veröffentlicht. Modelle wie das V-Modell XT, der RUP/SPEM oder HERMES geben IT-Unternehmen zur Beschreibung ihrer Entwicklungsprozesse eine Vielzahl erprobter Produkt-, Aktivitäts- und Rollenvorlagen sowie gut erforschte Prozessmuster an die Hand. Die Verwendung dieser standardisierten Prozess- und Vorgehensmodelle bietet drei wichtige Vorteile. So können Organisationen erstens auf einen großen Umfang vordefinierter Bausteine zurückgreifen und müssen somit nicht alles selbst neu entwerfen. Zweitens können sie ihre Konformität zu den Standards und damit eine entsprechende Prozessreife belegen, und drittens wird die organisationsübergreifende Zusammenarbeit durch standardisierte Produkt- und Verfahrensschnittstellen deutlich verbessert.

Die Anforderungen an Entwicklungsprozesse und die unternehmerischen Rahmenbedingungen jeder Organisation sind sehr individuell, so dass der Nutzen eines Vorgehensmodells für eine Organisation entscheidend von der Anpassbarkeit an ihre speziellen Bedürfnisse abhängt. Um dem Rechnung zu tragen, definieren Vorgehensmodellstandards wie das V-Modell XT (eXtreme Tailoring) einen Rahmen für detaillierte, organisationsspezifische Anpassungen. Vorgehensmodellvarianten, die in diesem vordefinierten Rahmen angepasst wurden, sind konform zum V-Modell XT und belegen damit eine entsprechende Reife. Bei der Entwicklung des V-Modell XT wird angestrebt, eine präzise und damit formale Beschreibung des Variantenraums vorzugeben. Dadurch wird es bspw. Werkzeugherstellern ermöglicht, Modellierungswerkzeuge zur organisationsspezifischen Anpassung anzubieten, die während der Modellierung die Konformität zum V-Modell XT jederzeit analysieren können.

Für die Beschreibung des Variantenraums eines Modells gibt es zwei grundlegend unterschiedliche Paradigmen. Einerseits besteht die Möglichkeit mit einem Metamodell einen Raum von Modellvarianten präzise zu beschreiben. Beispielsweise schränkt das V-Modell XT XSD-Metamodellschema die Menge der XML-Modelle ein, die dem V-Modell Metamodell entsprechen und trifft so eine Aussage darüber, ob ein XML-Modell im Variantenraum des V-Modells enthalten ist. Diese Art der Beschreibung identifiziert alle bei der Modellierung verwendbaren Modellelementtypen und enthält eine Menge von Bedingungen, die zwischen Instanzen dieser Typen gelten müssen. Durch die Analyse eines Modells kann nun beurteilt werden, ob dieses eine gültige Instanz eines bestimmten Metamodells ist. Diese Möglichkeit, den Variantenraum zu beschreiben, wird im Folgenden als analytische Variabilitätsbeschreibung bezeichnet.

Zur Formulierung analytischer Variabilitätsbeschreibungen existieren eine Reihe formaler sehr ausdrucksstarker Sprachen wie XSD, OCL oder Logiken (bspw. Prolog, basierend auf Prädikatenlogik). Weiterhin gibt es umfangreiche technologische Unterstützung wie Konsistenzchecker oder Inferenzmaschinen zur Analyse von formalen Modellbeschreibungen. Während der Vorgehensmodellanpassung haben die Prozessingenieure volle Flexibilität bei der Modelländerung, allerdings muss eine fertig gestellte Variante als gültig im Sinne der analytischen Variabilitätsbeschreibung nachgewiesen werden. Um so eine gültige Variante zu modellieren, muss der Prozessingenieur ein Verständnis für die analytisch formulierte und möglicherweise sehr umfangreiche Variabilitätsbeschreibung erwerben. Erst mit diesem Verständnis kann er gewünschte aber gleichzeitig Vorgehensmodell-konforme Modellanpassungen ableiten. Eine Auswahl von per se konformen Anpassungsmöglichkeiten steht ihm nicht als Unterstützung wie bei der konstruktiven Variabilitätsbeschreibung zur Verfügung.

Bei der konstruktiven Variabilitätsbeschreibung wird einerseits ein Referenzmodell und andererseits eine Reihe von Änderungsoperationen vorgegeben [Kuh08, Ter09]. Jede Änderungsoperation transformiert ein Modell in ein geändertes Modell. Durch die Anwendung einer beliebigen, endlichen Sequenz der vorgegebenen Änderungsoperationen auf das Basismodell können so genau die Modellvarianten erzeugt werden, die im Sinne dieser Beschreibung als gültig erachtet werden.

Diese Ausgangssituation zeigt, dass es bei der Beschreibung von Variabilität von Vorgehensmodellen ein Spannungsfeld zwischen der Verständlichkeit von Änderungsschritten (und damit deren Anwendbarkeit) und der Mächtigkeit der Anpasserunterstützung gibt. In diesem Papier wird der Unterschied der beiden oben informell formulierten Paradigmen zur Beschreibung von Variabilität in Abschnitt 3 präzise beschrieben, um daran in Abschnitt 4 die angedeuteten Probleme genau zu benennen und konkrete Herausforderungen für die Beschreibung der Variabilität von Vorgehensmodellen abzuleiten. In Abschnitt 2 werden aus dem wissenschaftlichen Umfeld einige Ansätze zur Bewältigung der identifizierten Herausforderungen erläutert.

2 Stand der Forschung

Da die Beschreibung individueller Anpassbarkeit großen Einfluss auf die Wartbarkeit und die Verständlichkeit von Software und Prozessbeschreibungen hat, gibt es im wissenschaftlichen Umfeld und in der Praxis unterschiedliche Ansätze zur Spezifizierung von Variabilität.

In objektorientierten Programmier- und Modellierungssprachen, bspw. bei der Gestaltung von Frameworks, wird das Konzept der Vererbung zur Beschreibung verschiedener Spezialisierungen mit gemeinsamen Eigenschaften häufig eingesetzt. Eine spezialisierte Klasse ist nicht nur durch ihre eigene Klassendefinition beschrieben, sondern auch durch die Klassendefinition ihrer Elternklasse(n). Ändert sich das Verhalten der Elternklasse, so beeinflusst dies das Verhalten all ihrer Spezialisierungen. Die Semantik der Vererbungsrelation wird meist durch eine Teilmengen-Beziehung (Spezialisierung $\subseteq$ Generalisierung) beschrieben, wobei die Relation meistens als transitiv aber nicht symmetrisch definiert ist (Zyklenfreiheit). Die Sicherung der Zyklenfreiheit wird in der Praxis durch Codeanalyse bzw. Modellvalidation erreicht und demzufolge dem analytischen Paradigma entspricht.

Im Gebiet von Vorgehensmodellen wurde das Spezialisierungskonzept im SPEM 2.0 [OMG08] eingesetzt. Eine Instanz des SPEM 2.0 Metamodells ist beispielsweise der Rational Unified Process (RUP). SPEM beschreibt vier sog. Variability Operations, mit denen definiert werden kann, dass ein Prozesselement ein Anderes erweitert (extends), ersetzt (replaces), ersetzt und dabei erweitert (extends-replaces) oder zu dessen Attributen ergänzende Werte beiträgt (contributes).

Die Variability Operation *extends* verhält sich vergleichbar der o.g. Spezialisierung, d.h. ein erweiterndes Prozesselement erwirbt Eigenschaften des Elternelements, ergänzt diese jedoch um spezifische Attribute und Beziehungen. *replaces* ist eine neue Form, die bei Programmiersprachen keinen Einzug genommen hat. Das Elternelement wird dabei vollständig durch das neue Element ersetzt, d.h. alle eingehenden und ausgehenden Beziehungen werden durch sein Kind übernommen. Dabei erwirbt das Kind jedoch keine Eigenschaften des Elternelements, denn dafür ist die Variability Operation *extends-replaces*.

Bei der Entwicklung der neuen Version 1.3 des V-Modell XT [TK09] wurde ein Erweiterungskonzept implementiert, mit dem die organisationsspezifische Anpassung des Vorgehensmodells unterstützt wird. In seiner derzeitigen Form erlaubt das Konzept zwar keine Spezialisierung im o.g. Sinne, dafür aber sog. Änderungsoperationen, die deklarativ Änderungen an Prozesselementen beschreiben. Es gibt Änderungsoperationen für das Umbenennen von Produkten, Rollen etc., sowie Änderungsoperationen, die Verantwortlichkeitsbeziehungen entfernen oder austauschen. Außerdem gibt es Operationen, mit denen sich an vorhandene Beschreibungstexte zusätzliche Textpassagen ergänzen lassen.

Die sichtbarste Verwandtschaft zu den Variability Operations zeigt sich bei diesen Änderungsoperationen zur Operation *contributes*. Beim SPEM wird das Ergänzen von Beziehungen damit realisiert.

Ein ganz anderer Ansatz, mit Variabilität umzugehen, sind Produktlinien [CN02]. Eine Produktlinie ist eine Menge von Produkten, die gemeinsame, abgegrenzte Merkmale besitzen. Jedes Produkt setzt sich aus einer Reihe von Merkmalen zusammen. Das Besondere an Produktlinien ist, dass alle Produkte in der Regel sehr viele gemeinsame Hauptmerkmale besitzen, sich im Einzelnen jedoch durch individuelle oder gruppenspezifische Merkmale voneinander unterscheiden. Auf diese Weise werden Produkte aus den zur Verfügung stehenden Merkmalen zusammengestellt. Damit das funktioniert, werden die Merkmale so konzipiert, dass sie möglichst voneinander losgelöst realisiert und über klare Schnittstellen miteinander kombiniert werden können. Im Gegensatz zu der Herangehensweise mit Spezialisierung, Variability Operations und Änderungsoperationen sind Produktlinien weniger auf Änderungen fokussiert, sondern auf die Konfiguration von bereits vorhandenen Merkmalen zu neuen Produkten. Der Variantenraum wird durch die im Vorfeld bekannten Merkmale aufgespannt.

3 Arten der Variabilitätsdefinition

Eine *Variabilitätsdefinition* beschreibt eine Menge von *gültigen* bzw. *konformen Varianten* eines ausgezeichneten Modells (sog. *Referenzmodell*). Im Allgemeinen enthält der Variantenraum eines Modells unendlich viele Varianten (bspw. schon allein durch Benutzung der Wertebereiche String oder Integer für diverse Attributwertbelegungen). Aus diesem Grund ist es nicht möglich alle Varianten explizit anzugeben. Allerdings muss eine Variantenraumdefinition endlich sein, um vom organisationsspezifischen Anpasser eines Vorgehensmodells überhaupt vollständig gelesen werden zu können. In diesem Abschnitt werden die beiden Paradigmen der analytischen und konstruktiven Variabilität zunächst vorgestellt, und anschließend hinsichtlich ihrer Definition und Nutzung formal erfasst.

3.1 Paradigma der analytischen Variabilitätsdefinition

Das analytische Paradigma basiert auf dem Denken in *Zuständen*. Eine analytische Beschreibung des Variantenraums beantwortet die Frage „welche Eigenschaften muss eine gültige Variante erfüllen?“. Völlig offen ist dagegen, wie ein Bearbeiter zu einer solchen Variante gelangt. Es werden keinerlei Einschränkungen hinsichtlich einzelner Bearbeitungsschritte gemacht (aber auch keine Hilfestellungen gegeben), so dass Zwischenversionen, die während der Bearbeitung entstehen, nicht der Variabilitätsbeschreibung entsprechen müssen. Ist die Bearbeitung abgeschlossen, so muss das Resultat gültig im Sinne der analytischen Beschreibung des Variantenraums sein, um als Variante dieses Raumes zu gelten.

3.2 Paradigma der konstruktiven Variabilitätsdefinition

Das konstruktive Paradigma basiert auf dem Denken in *Veränderungen*. Eine konstruktive Beschreibung des Variantenraums beantwortet die Frage „welche Änderungen (an einer gegebenen Referenz) können vorgenommen werden?“. Einem Bearbeiter wird zur Veränderung der Referenz nur die Anwendung vordefinierter Änderungsoperationen erlaubt. Im Gegenzug führt jede gültige Kombination von Änderungsoperationsanwendungen zu einer gültigen Variante – das Ergebnis muss nicht nochmals geprüft werden.

3.3 Formale Fundierung

Beide in 3.1 und 3.2 eingeführten Paradigmen sollen in diesem Abschnitt vor dem Hintergrund der Variabilität von Vorgehensmodellen präziser beschrieben werden. Diese Beschreibung bietet anschließend in Abschnitt 4 die Möglichkeit, anhand von Beispielen die besonderen Eigenschaften (Vor- und Nachteile) beider Paradigmen zu erläutern.

3.3.1 Analytische Variabilitätsdefinition

Eine analytische Variabilitätsdefinition $d_a \in D_{Analyt}$ kann als eine berechenbare Funktion f ausgedrückt werden, die für jedes Modell $m \in M$ nach endlicher Zeit ausgibt, ob es dem definierten Variantenraum entspricht.[1] Der so definierte Variantenraum selbst, also die konkrete, meist unendliche Menge zugehörender Modelle, wird als Extension von d_a bezeichnet – notiert als $Ext_{Analyt}(d_a)$.

D_{Analyt}	$D_{Analyt} = \{ d_a \in \{f: M \rightarrow Bool \mid\} \mid\}$
$Ext_{Analyt} : D_{Analyt} \rightarrow \wp(M)$	$Ext_{Analyt}(d_a) = \{ m \in M \mid d_a(m)=true \}$

Die Arbeit mit einer analytischen Variabilitätsdefinition stellt besondere Anforderungen an den organisationsspezifischen Anpasser eines Vorgehensmodells. So muss dieser durch ein gutes Verständnis der Variabilitätsbeschreibung abschätzen können, welche Änderungen er am Referenzmodell durchführen darf, damit das resultierende Modell noch konform zur vorgegebenen Variabilitätsdefinition ist. Fehlt ein ausreichendes Verständnis bspw. aufgrund einer sehr komplex beschriebenen Variabilitätsdefinition, so ist die Gefahr groß, dass die durch den Anpasser erstellte Variante nicht der Variabilitätsdefinition entspricht und wiederholt angepasst werden muss. Diese Prozedur kann langwierig und ineffizient sein, denn die erstellte Vorgehensmodellvariante muss sowohl den Anforderungen der Organisation als auch der vorgegebenen Variabilitätsdefinition entsprechen. Beispielsweise kann man ein XSD-Metamodellschema als Beschreibungsmittel für die geltenden Regeln einer Funktion d_a heranziehen. Anhand dieser Funktion kann dann analytisch entschieden werden, ob ein konkretes XML-Modell $x \in M$ der Variabilitätsbeschreibung entspricht ($d_a(x)=true$).

---

[1] D_{Analyt} ist die Menge aller analytischen Variabilitätsdefinitionen, M die Menge alle Modelle.

3.3.2 Konstruktive Variabilitätsdefinition

Eine konstruktive Variabilitätsdefinition $d_k \in D_{Kstrkt}$ besteht aus zwei Teilen:[2] Zunächst wird ein Referenzmodell $ref \in M$ vorgegeben, das bereits in dem zu definierenden Variantenraum liegt (es gilt $ref \in Ext_{Kstrkt}(d_k)$). Den zweiten Teil von d_k bildet eine endliche Menge von Änderungsoperationen $\{op_1..op_n\} \subseteq OP$, wobei jede Änderungsoperation eine Funktion aus folgender Menge ist:

$$OP=\{op: M \times PARAM \times .. \times PARAM \rightarrow M \mid\}$$

Dabei steht PARAM für die Menge möglicher Parameter die der Funktion zusätzlich zu einem Modell übergeben werden. Wendet man eine gültige Folge einzelner Änderungsoperationen rekursiv auf die Referenz an, so erhält man eine beliebige gültige Variante der konstruktiven Variabilitätsdefinition. Alle Wörter, die sich nicht mit einer beliebigen Abfolge von Änderungsoperationen erzeugen lassen, sind nach dieser Variabilitätsdefinition als nicht gültig anzusehen.

$$D_{Kstrkt} = \{ (ref,Op) \in M \times OP \}$$

$$Ext_{Kstrkt}(d_k) = \{ ref \} \cup U_{op \in Op}: Ext_{Kstrkt}((op(ref),Op))$$

Die Arbeit mit einer konstruktiven Variabilitätsdefinition erspart dem organisationsspezifischen Anpasser eines Vorgehensmodells ein umfassend detailliertes Verständnis der gesamten Variabilitätsdefinition besitzen zu müssen. Da jede Anwendung einer Änderungsoperation zu einer konformen Variante führt, muss der Anpasser lediglich die geeigneten Operationen wählen (und ggf. deren Parameter mit passenden Argumenten befüllen – um bspw. das Element auszuwählen an dem eine Änderung vorgenommen werden soll). Die Herausforderung beim Design einer konstruktiven Variabilitätsdefinition d_k besteht darin, zu einem Referenzmodell d_k.ref eine Menge von Änderungsoperationen zu definieren, die in beliebiger rekursiver Anwendung genau die Modelle erzeugen, die als konforme Varianten auch tatsächlich vorgesehen sind. Andererseits soll die Semantik der Änderungsoperationen auch intuitiv und von der Anzahl überschaubar für Anpasser des Modells sein.

4 Einfluss der Variabilitätsdefinition auf die Modellanpassung

Je nach Komplexität der Variabilitätsbeschreibung und Erfahrung des Modellierers ist eine analytische bzw. konstruktive Definition von Variabilität vorteilhafter zur organisationsspezifischen Anpassung von Vorgehensmodellen. In diesem Abschnitt sollen für beide Paradigmen Modellierungsbeispiele aufgezeigt werden, die die Vor- und Nachteile beider Ansätze illustrieren. So werden im folgenden Abschnitt die Einschränkungen der konstruktiven Variabilitätsdefinition aufgezeigt bevor anschließend die Schwächen der analytischen Variabilitätsdefinition betrachtet werden.

---

[2] D_{Kstrkt} ist die Menge aller konstruktiven Variabilitätsdefinitionen.

4.1 Probleme bei der konstruktiven Variabilitätsdefinition

Für das folgende Beispiel soll sowohl ein Auszug aus einer analytischen als auch aus einer konstruktiven Definition von Variabilität angegeben werden, die beide denselben Variabilitätsraum beschreiben. Anhand der verwendeten Paradigmen zur Variabilitätsdefinition sollen die Probleme einer *konstruktiven* Definition herausgehoben werden.

Das für das Szenario exemplarisch betrachtete Vorgehensmodell beschreibt die Bearbeitungs- und Prüfbeziehungen zwischen Rollen und Produkten unter Beachtung des Vier-Augen-Prinzips. Wie in Abbildung 1 (links) dargestellt, erarbeiten laut Referenzmodell der Projektleiter und der QS-Verantwortliche u.a. die beiden Produkte Projektabschlussbericht und Abnahme-Erklärung, und prüfen diese Dokumente gegenseitig.

Zum Variabilitätsraum sollen nun solche Varianten gezählt werden, die ebenfalls das Vier-Augen-Prinzip betrachten, jedoch andere Zuordnungen der Erarbeitungs- und Prüfaufgaben besitzen. Diesen Sachverhalt will eine Organisation bei der Erstellung ihres spezifischen Vorgehensmodells nutzen, so dass einige Erarbeitungs- und Prüfbeziehungen jeweils vertauscht werden sollen (Abb.1 rechts). Ein Grund dafür könnte bspw. sein, dass diese Modellierung besser zu den schon bestehenden Strukturen in der Organisation passt.

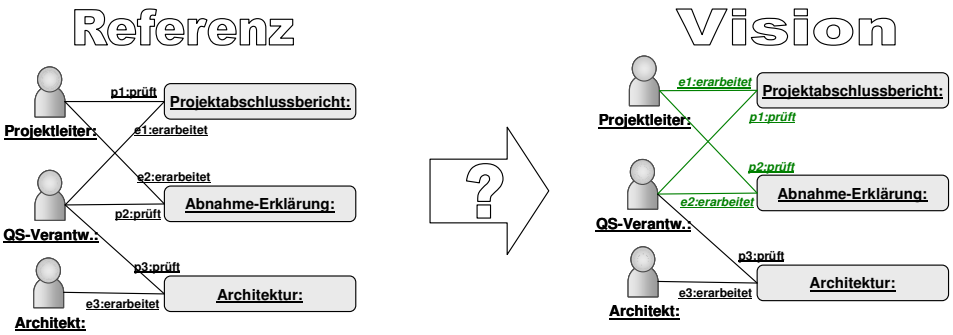


Abbildung 1: Gewünschte Ableitung einer Variante aus dem Referenzmodell

Nach der analytischen Variabilitätsdefinition kann eine Organisation die gewünschte Version direkt erzeugen, muss jedoch prüfen ob ihr angepasstes Vorgehensmodell auch eine gültige Variante des vorgegebenen Referenzmodells ist.

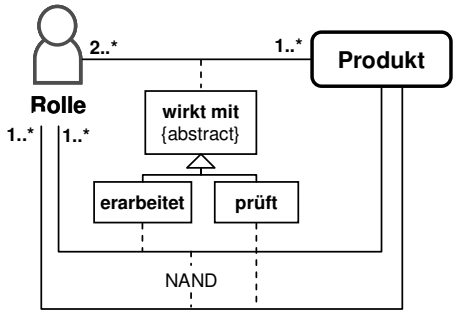


Abbildung 2: Analytische Definition des Variabilitätsraums

Abb. 2 zeigt die in diesem Szenario geltende, analytische Variabilitätsdefinition in UML-Syntax. Diese Definition ist kompakt und relativ eingängig. So beschreibt sie, dass jede Rolle mindestens an einem Produkt mitwirken muss, und dass ein Produkt mindestens durch eine Person erarbeitet und eine weitere geprüft werden muss, sodass Bearbeiter und Prüfer nicht dieselbe Person sein dürfen. Anders formuliert steckt hinter dieser Definition genau das Vier-Augen-Prinzip für die Bearbeitung bzw. Prüfung von Dokumenten und die Anforderung dass das Modell keine Rollen enthalten soll, die an keinem Produkt mitwirken, und keine Produkte, die durch niemanden bearbeitet bzw. geprüft werden. Sowohl das Referenzmodell als auch die gewünschte Variante (Abb.1 links und rechts) erfüllen diese Eigenschaften. Die gewünschte organisationsspezifische Variante ist also gültig.

Im Gegensatz zur analytischen ist die Anpassung auf Basis einer konstruktiven Variabilitätsdefinition wesentlich unvorteilhafter.

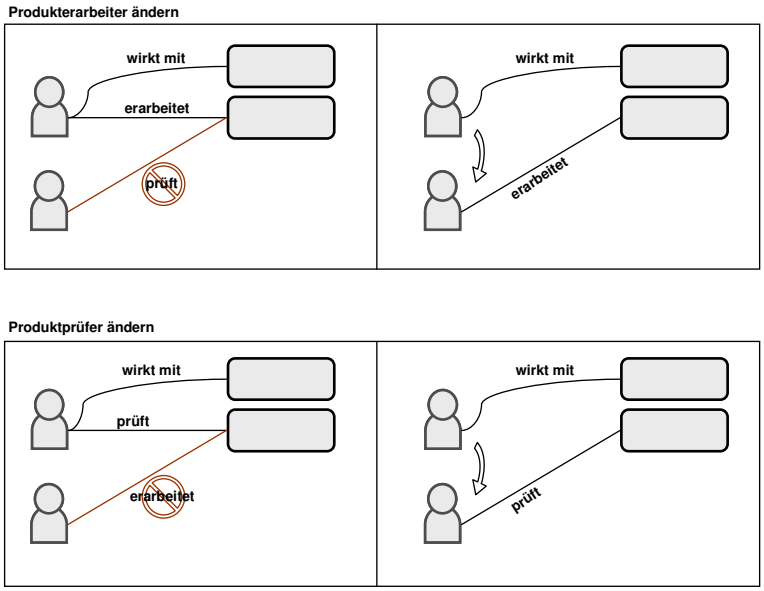


Abbildung 3: Konstruktive Definition des Variabilitätsraums

In Abb. 3 werden beispielhaft zwei Änderungsoperationen aus einer äquivalenten, konstruktiven Variabilitätsbeschreibung dargestellt. Die linke Seite beschreibt darin ein Suchmuster (die zu ersetzende Modellstruktur) und die rechte Seite beschreibt die ersetzte Modellstruktur. So beschreibt die erste Änderungsoperation „Produkterarbeiter ändern“ die Ersetzung der Erarbeitungsbeziehung sofern der bisherige Bearbeiter noch an anderen Produkten mitwirkt und das betreffende Produkt durch den neuen Erarbeiter nicht schon geprüft wird. Dies garantiert die oben geforderten Strukturen (Vier-Augen-Prinzip usw.). Die zweite Änderungsoperation „Produktprüfer ändern“ beschreibt analog dazu die Ersetzung der Prüfbeziehung. Diese beiden Änderungsoperationen mit dem Referenzmodell aus Abbildung 1 reichen noch nicht aus, um alle Varianten der analytischen Definition aus Abbildung 2 zu konstruieren. Allerdings sind die beiden Operationen ausreichend, um die gewünschte organisationsspezifische Variante (Abb. 1 rechts) aus dem Referenzmodell zu erzeugen.

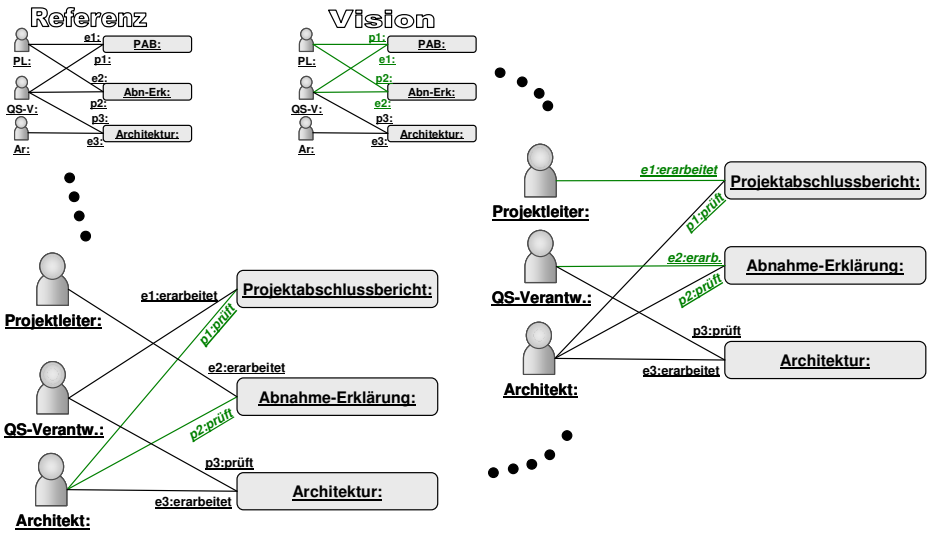


Abbildung 4: Der konstruktive, schrittweise Ableitungspfad zur gewünschten Variante

Abbildung 4 zeigt einen möglichen Ableitungspfad vom Referenzmodell zur Variante auf Basis der beiden in Abbildung 3 dargestellten Änderungsoperationen. So werden im ersten Schritt durch zweimalige Anwendung der zweiten Änderungsoperation die beiden Prüfbeziehungen „temporär“ auf den Architekten „umgebogen“. Im zweiten Schritt werden durch zweimaliges Anwenden der ersten Änderungsoperation die Bearbeitungsbeziehungen, wie in der Variante gewünscht, geändert bevor abschließend die Prüfbeziehungen auch, so wie in der Variante gewünscht, geändert werden können.

Diese Art der Anwendung der Änderungsoperationen wirkt zu recht sehr umständlich und als Anpasser des Modells in diesem Szenario wünscht man sich triviale Lösch- bzw. Hinzufügeoperationen für die beiden Mitwirkungsbeziehungen. Würde man in der konstruktiven Variabilitätsbeschreibung diese zusätzlich zur Verfügung stellen, so enthielte der Variantenraum allerdings auch Varianten, in denen Bearbeiter und Prüfer eines Produktes identisch sind (Verletzung des Vier-Augen-Prinzips), oder Vorgehensmodelle mit Rollen, die an keinem Produkt mitwirken und Produkte die durch niemanden erarbeitet und/oder geprüft werden.

Schon dieses sehr einfach gestaltete Szenario zeigt, dass es sehr schwierig bzw. unmöglich ist, den Variantenraum konstruktiv durch Referenzmodell und Änderungsoperationen zu beschreiben, so dass die vorgegebenen Operationen intuitiv durch einen Modellanpasser anwendbar sind. Der gewonnene Vorteil, dass durch Anwendung der vordefinierten Änderungsoperationen der Bearbeiter sich nicht um die Konformität der erzeugten Variante kümmern muss, ist damit von zweifelhaftem Nutzen.

4.2 Probleme bei der analytischen Variabilitätsdefinition

Im vorherigen Abschnitt 4.1 wurde illustriert, dass die Einschränkung der Modellbearbeitung durch eine genau definierte Auswahl von Änderungsoperationen einer konstruktiven Variabilitätsbeschreibung die Modellierung sehr umständlich macht. In diesem Abschnitt wird ein Beispiel vorgestellt, bei dem die Vorgabe von Änderungsoperationen zur Definition der Variabilität und zur direkten Benutzung durch die Modellierer, die Einhaltung der Konformität gegenüber einer analytischen Beschreibung deutlich erleichtert.

Das Beispiel thematisiert die Dekomposition und Integration eines zu entwickelnden Systems wie sie in der V-Darstellung der meisten Vorgehensmodellen im Software Engineering beschrieben ist. Als Beispiel dafür sei in Abbildung 5 ein Ausschnitt aus dem V-Modell XT dargestellt

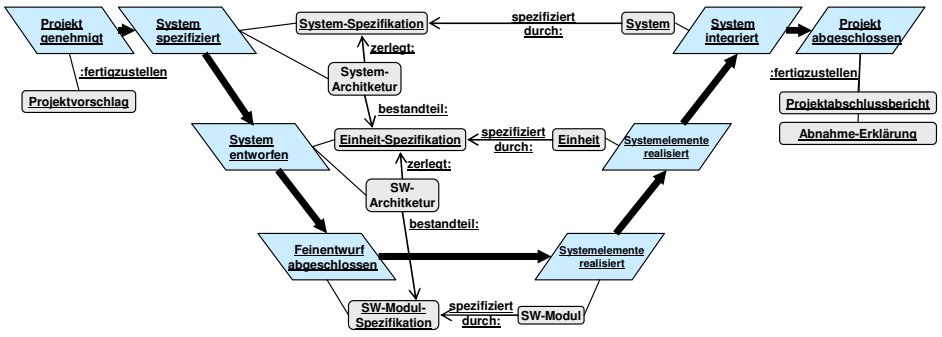


Abbildung 5: Vereinfachte Darstellung der Dekomposition und Integration im V-Modell

In Abbildung 5 sind auf dem linken Schenkel des Vs die Entscheidungspunkte (Quality Gates) der Systemdekomposition zu sehen. Zu jedem dieser Entscheidungspunkte muss ein Produkt mit der jeweiligen Spezifikation (System, Einheit, Modul) fertig gestellt sein. Soll die jeweilige Spezifikation nach einem Entscheidungspunkt weiter zerlegt werden, muss zusätzlich ein Produkt mit der Beschreibung der jeweiligen Architektur erstellt werden. Sind die jeweils feingliedrigsten Systemelemente realisiert, wird anhand der jeweils übergeordneten Architektur das nächstgrößere Systemelement integriert (rechte Seite des Vs).

Ausgehend von der Berücksichtigung der Zerlegung eines Systems in einem Vorgehensmodell, soll im Folgenden eine konstruktive Variabilitätsbeschreibung des Vs angegeben werden. Konkret soll für dieses Szenario die Anzahl der Zerlegungsebenen in den Varianten eines Vorgehensmodells variabel sein. Als Startpunkt sei das in Abbildung 6 dargestellte Referenzmodell vorgegeben.

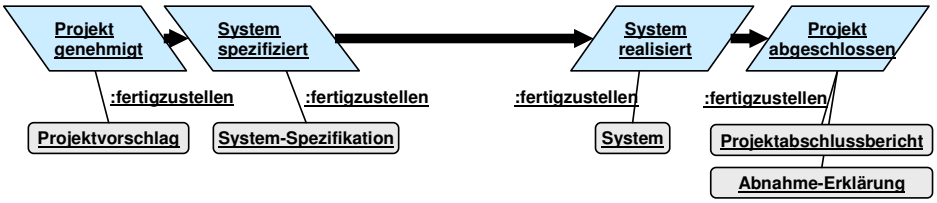


Abbildung 6: Referenzmodell der konstruktive Variabilitätsdefiniton

In diesem Referenzmodell gibt es nur eine Ebene bzgl. Dekomposition/Integration des Systems, so dass es vor der Realisierung nicht weiter zerlegt und integriert wird. Allerdings soll es in diesem Szenario zugelassen sein, dass das Modell um beliebige Dekompositions-/Integrationsebenen erweitert werden kann, um bspw. eine Struktur wie im V-Modell XT (siehe Abb. 5) erzeugen zu können. Dies kann durch die Einführung einer Änderungsoperation „Zerlegungsebene hinzufügen“ erreicht werden (siehe Abbildung 7).

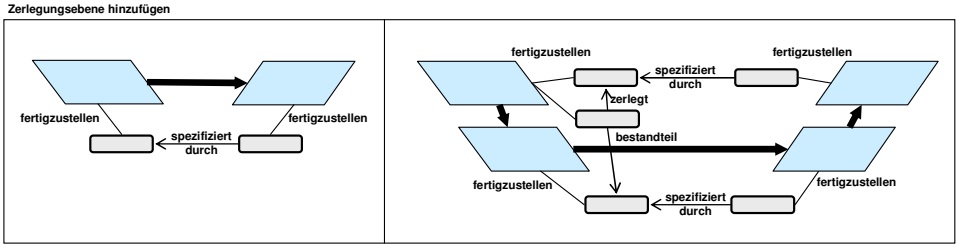


Abbildung 7: Änderungsoperation der konstruktiven Variabilitätsdefinition

Das Suchmuster auf der linken Seite der Abbildung stellt die zu ersetzende Struktur dar und das Ersetzungsmuster auf der rechten Seite enthält die Entscheidungspunkte, Produkte und Beziehungen zur Einführung einer weiteren Ebene. In Abbildung 8 ist exemplarisch eine Anwendung dieser Änderungsoperation dargestellt. Im letzten Schritt der Modelländerung werden die neu geschaffenen Entscheidungspunkte und Produkte lediglich benannt.

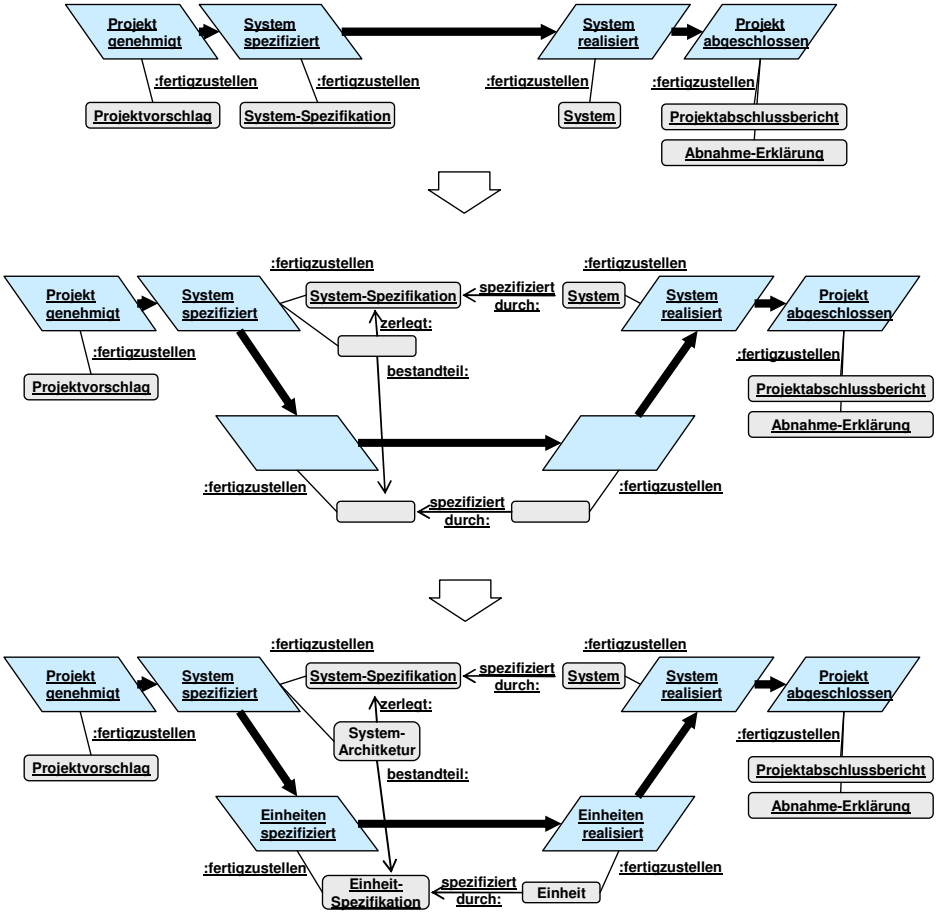
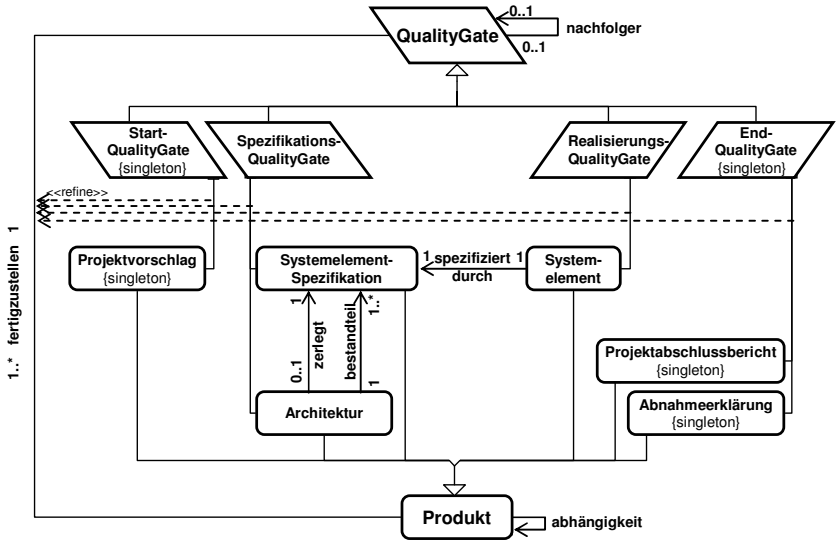


Abbildung 8: Anwendungsbeispiel der Änderungsoperationen der konstruktiven Variabilitätsbeschreibung

Durch die Beschränkung auf die Anwendung dieser Änderungsoperation für die Modellbearbeitung wird eine Vielzahl an Bedingungen, die an die Struktur der Varianten gestellt wird abgesichert. So können Spezifikations- oder Architekturdokumente weder ausgelassen noch zu einem unpassenden QualityGate zugeordnet werden. Insbesondere kann der v-förmige Ablauf generell nicht in seiner Struktur verändert werden: zunächst muss von der obersten Granularitätsstufe hin zur kleinsten spezifiziert und anschließend im umgekehrten Detaillierungsgrad integriert werden.

All diese beschriebenen Bedingungen, die an die Dekompositions-/Integrationsstruktur einer Variante gestellt werden, könnte man nur sehr komplex in einer analytischen Variabilitätsdefinition beschreiben. Für Vorgehensmodellvarianten mit beliebig vielen Dekompositions-/Integrationsebenen ist in Abb. 9 eine entsprechende analytische Beschreibung in UML-Syntax (angereichert mit OCL-Invarianten) abgebildet.




```
//Es gibt genau einen Anfang und genau ein Ende:
//- nur EndQualityGate hat keinen Nachfolger
context EndQualityGate inv: nachfolger.isEmpty()
context QualityGate inv: oclIsTypeOf(EndQualityGate) == false implies nachfolger.notEmpty()
//- nur StartQualityGate hat keinen Vorgänger
context QualityGate inv: nachfolger.forAll( n | n.oclIsTypeOf(StartQualityGate) == false )
context QualityGate inv: oclIsTypeOf(StartQualityGate) implies exists( q : QualityGate | q.nachfolger.contains( self ) )
//Verknüpfungsbeschränkungen:
//-Start hat genau ein SpezifikationsQualityGate als Nachfolger
context StartQualityGate inv: nachfolger.forAll( n | n.oclIsTypeOf(SpezQualityGate) == false )
//-SpezifikationsQualityGate hat entweder ein RealisierungsQualityGate oder ein SpezifikationsQualityGate als Nachfolger
context SpezQualityGate inv: nachfolger.forAll( n | n.oclIsTypeOf(SpezQualityGate) == true or n.oclIsTypeOf(RealQualityGate) )
//-RealisierungsQualityGate hat entweder ein RealisierungsQualityGate oder ein EndQualityGate als Nachfolger
context RealQualityGate inv: nachfolger.forAll( n | n.oclIsTypeOf(RealQualityGate) == true or n.oclIsTypeOf(EndQualityGate) )

//Die Anzahl von Spezifikations- und RealisierungsQualityGates muss gleich sein
context Vorgehensmodell inv: qualitygates.count(SpezQualityGate)=qualitygates.count(RealQualityGate)
//Die Bestandteile einer Architektur müssen untergeordnete Elemente sein
context Architektur inv: bestandteile->forAll( b | b!=zerlegt )
//Die Bestandteile müssen in einer untergeordneten Phase ausgestaltet werden
context Architektur inv: bestandteile->forAll( b | self.zerlegt.nachfolger = b.fertigzustellen )
```

Abbildung 9: analytische Variabilitätsbeschreibung für v-förmige Vorgehensmodelle

Anhand der in Abbildung 9 dargestellten, analytische Variabilitätsdefinition ist es sehr schwierig, Änderungen am Referenzmodell abzuleiten, die zu einer gültigen Variante führen, da eine schier unüberschaubare Anzahl an Bedingungen eingehalten werden muss. Somit ist es damit außerordentlich mühselig eine gültige Variante des Referenzmodells zu erzeugen, welche durch die vergleichsweise einfache Anwendung der oben beschriebenen Änderungsoperation ohne Weiteres und unter Berücksichtigung aller Bedingungen erzeugt werden kann.

5 Zusammenfassung und Ausblick

In diesem Beitrag wurden die Eigenschaften zwei wesentlicher Paradigmen zur Definition von Variabilität von Vorgehensmodellen herausgearbeitet. Ein besonderer Schwerpunkt wurde auf die Anwendbarkeit der Variabilitätsbeschreibung zur konformen Anpassung eines organisationsspezifischen Vorgehensmodells gelegt. Gerade bei der Einführung von Vorgehensmodellen ist die flexible Anpassung auf die Bedürfnisse von Organisationen außerordentlich wichtig, um die Nutzung von schon vorhandenem Prozesswissen und die Akzeptanz aller am Prozess beteiligter Akteure zu gewährleisten. Gleichzeitig wollen Unternehmen eine bestimmte Prozessreife gegenüber ihren Auftraggebern und Kunden nachweisen. Eine Möglichkeit diesen Nachweis zu erbringen, ist die konforme Anpassung eines standardisierten Vorgehensmodells, dass eine bestimmte Prozessreife garantiert und damit die Erstellung einer Variante, welche die gleichen Qualitätseigenschaften aufweist. Eine geeignete Beschreibung der Variabilität eines Vorgehensmodells stellt in der Praxis allerdings ein gewichtiges Problem dar. So wird i.d.R. einer von zwei grundlegenden Ansätzen zur Variabilitätsbeschreibung verfolgt. Einerseits kann im Rahmen einer analytischen Variabilitätsdefinition eine Menge von Invarianten beschrieben werden (Metamodell, XSD-Schema, OCL etc.), die die Strukturen möglicher Modellvarianten einschränken. Andererseits kann ein Referenzmodell mit einer Menge von erlaubten Änderungsoperationen vorgegeben werden, so dass sich jede Variante aus der rekursiven Anwendung der beliebig vieler, vorgegebener Änderungsoperationen ergibt. Je nach Anwendungsszenario haben beide Arten der Variabilitätsdefinition unterschiedliche Stärken und Schwächen, die in Abschnitt 4 ausführlich erläutert wurden.

Für das zukünftige Design von Variantenräumen von Vorgehensmodellen wäre eine kombinierte Variabilitätsbeschreibung, die sich die positiven Eigenschaften beider beschriebener Paradigmen zu Nutze macht, ein viel versprechender Ansatz. Dazu könnten eine Reihe von typischen Änderungsoperationen bzgl. eines Referenzmodells vorgegeben werden. Arbeitet ein Modellierer nur mit diesen Operationen, dann läuft er nicht Gefahr, die Konformität zum Standard zu verletzen. Zusätzlich dazu könnten dem Modellierer aber weitere Änderungsmöglichkeiten, die nicht mit expliziten Operationen vordefiniert sind, angeboten werden. Der Rahmen für diese zusätzlichen Modelländerungen kann dann durch eine analytische Beschreibung der Varianteneigenschaften beliebig eng gefasst werden, solange dieser die Anwendung der explizit angegebenen Änderungsoperationen nicht beschränkt. Die Definition und Anwendung (formaler) Variabilitätsbeschreibungen von Vorgehensmodellen steht in der wissenschaftlichen und industriellen Praxis aufgrund der wenigen Vorarbeiten noch ganz am Anfang obwohl das Problembewusstsein diesbezüglich hoch ist.

Literaturverzeichnis

[CN02] P. Clements and L. Northrop, *Software Product Lines: Practices and Patterns*. Addison-Wesley Professional, Aug 2002.

[TK09] T. Ternité and M. Kuhrmann, "Das VModell XT 1.3 Metamodell," Technische Universität München, Forschungsbericht TUMI0905, Mar. 2009.

[OMG08] Object Management Group (OMG), "Software Process Engineering Meta-Model, version 2.0," Apr 2008. [Online]. Available: http://www.omg.org/technology/documents/formal/spem.htm

[Kuh08] M. Kuhrmann, "Konstruktion modularer Vorgehensmodelle", Dissertation, Technische Universität München, 2008

[Ter09] T. Ternité, "Process lines: a product line approach designed for process model development", 35th EUROMICRO Conference on Software Engineering and Advanced Applications, Track SPPI, in Erscheinung, 2009.