

Modellierung von Variabilität mit UML Use Cases

Thomas von der Maßen, Horst Lichter
{vdmass, lichter}@cs.rwth-aachen.de
Lehr- und Forschungsgebiet Informatik III
RWTH Aachen

1 Einführung

Die Entwicklung einer Software-Produktlinie stellt hohe Ansprüche an den gesamten Softwareprozess, insbesondere auch an das Requirements Engineering. Clements und Northrop definieren eine Produktlinie¹ wie folgt [Cle01]: “A software product line is a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission”. Eine Produktlinie besteht demnach aus einer Menge von Produkten, die auf Basis einer Plattform entwickelt werden, die die gemeinsamen Eigenschaften aller Produkte realisiert. Die einzelnen Produkte unterscheiden sich bei einer Reihe von Eigenschaften, dadurch dass diese gar nicht oder unterschiedlich ausgeprägt sind. Die Identifikation und Modellierung von gemeinsamen und variablen Eigenschaften ist eine essentielle Aufgabe, weil diese kommuniziert und abgestimmt werden müssen und die Entwicklung einer Produktlinie maßgeblich beeinflussen.

2 Modellierung von Variabilität

David M. Weiss definiert Variabilität wie folgt [Wei99]: „An assumption about how members of a family may differ from each other.“ Variable Merkmale unterscheiden somit die Produkte einer Produktlinie; sie liegen in unterschiedlichen Ausprägungen vor. Die variablen Merkmale einer Produktlinie werden mit Hilfe von Variationspunkten modelliert, die insbesondere auch die zur Verfügung stehenden Varianten definieren. Durch Auswahl einer bereitgestellten Variante wird ein Variationspunkt aufgelöst.

In der Literatur [z.B. Eis00] werden die folgenden Variabilitätstypen unterschieden, die an Variationspunkten auftreten können:

- *Optionen* können ausgewählt bzw. berücksichtigt werden oder nicht. Sie stellen somit eine 0 oder 1 Auswahl dar.

- Aus einer Menge von *Alternativen* muss genau eine Alternative ausgewählt werden. Sie entspricht somit einer 1 aus N Auswahl.
- *Optionale Alternativen* bilden eine Kombination aus einer Option und mehreren Alternativen, wobei entschieden werden kann, ob die Alternativen überhaupt berücksichtigt werden. Dies entspricht einer 0 oder 1 aus N Auswahl.

Variable Eigenschaften können unterschiedlichen Bereichen (z.B. Technik, Funktionalität) zugeordnet werden [Hal02] und treten in unterschiedlichen Modellierungsebenen auf. Wir unterscheiden zwischen

- Domänenvariabilität und
- Laufzeitvariabilität.

Während die Domänenvariabilität die verschiedenen Produktausprägungen beschreibt, beschreibt die Laufzeitvariabilität die unterschiedlichen Abläufe und Interaktionen zur Laufzeit [Maß02].

Eine weitere Dimension der Variabilität bilden die Variabilität der Produktstruktur (statische Sicht) und die Variabilität in den zu realisierenden Abläufen (verhaltensorientierte Sicht). In der statischen Sicht werden die gemeinsamen und die variablen Merkmale der Produktlinie modelliert. Hierzu können Feature-Diagramme [Eis00, Kan00] benutzt werden, bei denen die Merkmale im Wesentlichen hierarchisch entlang der Besteht-Aus-Beziehung strukturiert werden. Eine Notation, um Variabilität aus der verhaltensorientierten Sicht zu modellieren, fehlt hingegen.

3 Variabilität in Use Cases

Um variable Abläufe zu modellieren, bieten sich Use Cases und Use Case Diagramme an, da sie ein System ablauforientiert beschreiben und eine weite Akzeptanz im industriellen Umfeld genießen. Allerdings fehlen wesentliche Modellierungselemente, um Variabilität in Use Cases ausdrücken zu können. Abbildung 1 zeigt die Erweiterung des Meta-Modells für Use Cases. Die neu eingebrachten Modellierungselemente sind grau unterlegt. Sie werden im Folgenden beschrieben.

¹ Der zum Begriff Produktfamilien wird häufig synonym verwendet.

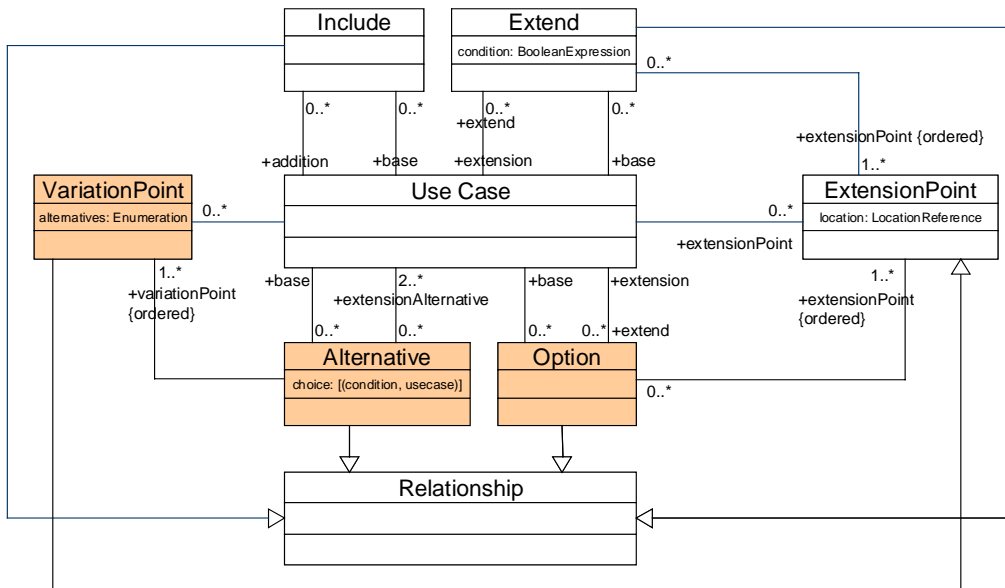


Abb 1: Erweiterung des Use Case Metamodells

Die *Option-Beziehung*, die zwischen einem Basis Use Case und einem optionalen erweiterten Use Case besteht, drückt aus, dass der optionale erweiterte Use Case in den Basis Use Case eingefügt werden kann. Die Ausführung hängt von keinerlei Bedingung ab. Die Stelle an der optionale erweiterte Use Case eingefügt werden kann, wird durch einen *Extension Point* angegeben.

Die *Alternative-Beziehung*, die zwischen einem Basis Use Case und mindestens zwei alternativen Use Cases besteht, drückt aus, dass genau ein alternativer Use Case innerhalb des Basis Use Cases ausgeführt werden muss. Da mehrere alternative Abläufe innerhalb eines Basis Use Cases auftreten können, müssen die zueinander alternativen Use Cases gruppiert werden können. Dazu dient das Modellierungselement *VariationPoint*, welches einen speziellen Erweiterungspunkt darstellt, indem er die zu einem Variationspunkt gehörende alternativen Use Cases auflistet.

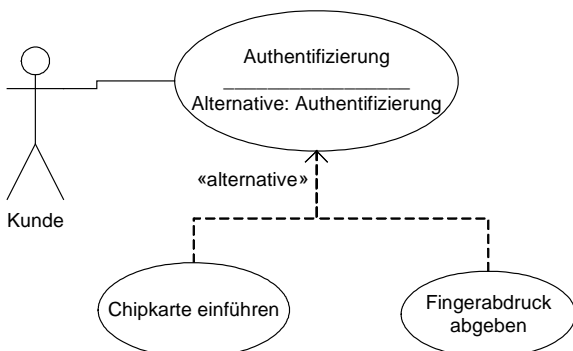


Abb 2: Alternative Use Cases

Die Abbildungen 2 und 3 zeigen im Kontext einer Produktlinie für Bankautomaten, wie die neuen Modellierungselemente genutzt werden können, um Domänenvariabilität zu modellieren: Abbildung 2 zeigt ein Use Case Diagramm für die Authentifizierung am Bankautomaten. Ein Kunde kann sich auf zwei alternative Weisen authentifizieren, abhängig von der Funktionalität, die ein Bankautomat bereitstellt. Verfügt der Automat über einen Chipkartenleser, so muss er den Use Case „Chipkarte einführen“ ausführen. Verfügt der Automat hingegen über einen Fingerabdruckscanner, so wird der Use Case „Fingerabdruck abgeben“ durchlaufen.

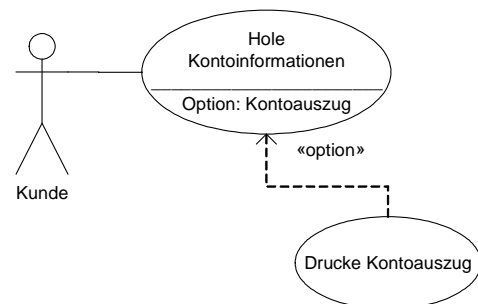


Abb 3: Optionaler Use Case

Abbildung 3 zeigt, dass der Use Case „Kontoauszug drucken“ optional ausgeführt werden kann. Auch diese Auswahl hängt allein von der bereitgestellten Produktfunktionalität ab, d.h., ob der Automat über einen Druckmechanismus verfügt oder nicht.

Die hier modellierte Domänenvariabilität zur Beschreibung unterschiedlicher Produktfunktionali-

tät wird aufgelöst, wenn die Funktionalität eines einzelnen Produkts festgelegt wird. Dazu kann eine optionale Beziehung entweder gänzlich entfallen, wenn das Produkt die optionale Funktionalität nicht unterstützt oder aber in eine extend- bzw. include-Beziehung transformiert werden, in Abhängigkeit davon, ob der vormals optionale Use Case nur unter bestimmten Bedingungen oder immer ausgeführt wird. Für eine Alternative-Beziehung muss entschieden werden, welche Produktfunktionalität bereitgestellt werden soll und der entsprechende alternative Use Case wird mit einer include-Beziehung mit dem Basis Use Case verbunden.

Die hier beschriebenen neuen Modellierungselemente für Use Cases können aber auch verwendet werden, um Laufzeitvariabilität auszudrücken. Abbildung 4 zeigt ein Beispiel hierfür: der Kunde muss sich bei Ausführung einer Transaktion entscheiden, ob er eine Überweisung tätigen oder einen Dauerauftrag einrichten möchte. Auch dieser Sachverhalt kann mit Hilfe einer Alternative-Beziehung ausgedrückt werden.

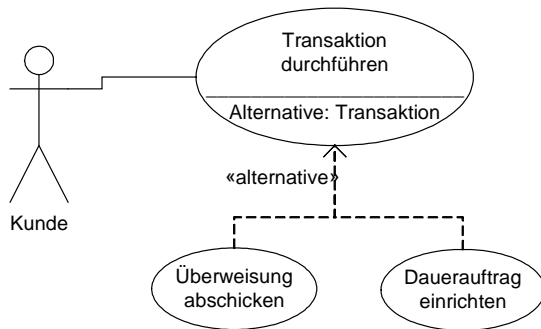


Abb 4: Modellierung von Laufzeitvariabilität

Die neu eingeführten Modellierungselemente lassen sich eins zu eins auch in eine natürlichsprachliche Use Case Beschreibung übernehmen

Name	Authentifizierung	
Ziel	Kunde erhält Zugang zu Kontoservices	
...	...	
Standardablauf	Schritt	Aktion
	1	System fordert Kunden zur Authentifizierung auf
	2	Kunde authentifiziert sich durch <Alt-Auth>
	3	System zeigt Hauptmenü an
Fehlerfälle	Schritt	Aktion
	2a	<Authentifizierung nicht erfolgreich> Fehlermeldung anzeigen
Variationen	Schritt	Aktion
	<Alt-Aut>	<alternative> <a1 cond = „Automat verfügt über Chipkartenleser“> UC „Chipkarte einführen“ ausführen-</a1> <a2 cond = „Automat verfügt über Fingerabdruckscanner“> UC „Fingerabdruck abgeben“ ausführen-</a2> </alternative>

Abb. 5 Erweiterte Use Case Beschreibung

Dazu kann eine Use Case Vorlage um eine neue Kategorie „Variationen“ erweitert werden, in der sämtliche variablen Abläufe eingefügt und beschrieben werden bzw. in der auf die alternativen und optionalen Use Cases verwiesen wird.

4 Zusammenfassung

Die Modellierung von Variabilität hat gerade, aber nicht ausschließlich, durch die Produktlinienentwicklung eine große Bedeutung gewonnen. Die notwendige Repräsentation von Variabilität dient zur Verbesserung des Verständnisses von komplexen Domänen, Prozessen und Systemen. Um Variabilität in seiner Gesamtheit zu identifizieren und zu kommunizieren, bedarf es der Modellierung auf verschiedenen Ebenen. Zudem ist es notwendig, neben der statischen Sicht, auch eine dynamische, verhaltensorientierte Sicht zu berücksichtigen, um variable Abläufe und Interaktionen zu identifizieren. Zusätzlich müssen unterschiedliche Typen von Variabilität ausgedrückt werden können. Wir haben die UML Use Cases dahingehend untersucht, in wie weit sie in der Lage sind Variabilität auszudrücken, mit dem Ergebnis, dass sie dazu nur unzureichend in der Lage sind. Um diesen Unzulänglichkeiten zu begegnen, haben wir eine Erweiterung des Metamodells vorgeschlagen. Mit Hilfe der neuen Elemente kann Variabilität in Use Cases ausdrucksstark und einfach modelliert werden, was die Kommunikation zwischen Entwicklern und Domänenexperten erleichtert.

Literatur

- [Cle01] Paul Clements, Linda M. Northrop. *A Framework for Product Line Practice - Version 3.0*. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2001. Verfügbar unter: www.sei.cmu.edu/plp/framework.html.
- [Eis00] Ulrich Eisenecker, Krzysztof Czarnecki. *Generative Programming – Methods, Tools, and Applications*. Addison-Wesley 2000.
- [Hal02] Günter Halmans, Klaus Pohl. *Modellierung der Variabilität einer Software-Produktfamilie*, Proceedings Modellierung 2002, p. 63-74. Lecture Notes in Informatics, Volume P-12, 2002.
- [Kan00] Kyo Kang et al. *Feature-Oriented Domain Analysis (FODA) Feasibility Study*. Technical report CMU/SEI-90-TR-021, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 2000.
- [Maß02] T. von der Maßen, H. Lichter, *Modeling Variability by UML Use Case Diagrams*, Proc. Int. Workshop on Requirements Engineering for Product Lines, Technical Report: ALR-2002-033.
- [Wei99] D. Weiss, C.T. Robert Lai, *Software Product-Line Engineering — A Family-Based Software Development Process*, Addison-Wesley, 1999.