

Slide content:

Top left: SysML/DSL für Anforderungen mit Varianten

Top right: oose. Innovative Informatik, b+m engineering

Main title: SysML/DSL für Anforderungen mit Varianten

Subtitle: REConf 2009, München

Speaker 1: Tim Weilkiens, Bereichsleiter Systems-Engineering, tim.weilkiens@oose.de

Speaker 2: Thomas Stahl, Chefarchitekt, t.stahl@bmiag.de

Wesentliche Techniken zum Umgang mit Varianten haben ihren Ursprung im Produktlinien-Engineering.

Die Variabilitäten der möglichen Einzelprodukte einer Produktlinie werden dabei systematisch erfasst und einer formalen Modellierung zugänglich gemacht. Die resultierenden Spezifikationen liegen dann in einer problemorientierten und nicht in einer lösungsorientierten Sprache vor.

Diese etablierten Techniken lassen sich hervorragend im Requirements Engineering einsetzen, um dort Varianten handhabbar zu machen - auch für Projekte, die sich nicht im Kontext einer Produktlinie befinden.

Der Vortrag zeigt Ihnen, wie SysML mit einem Profil bzw. einer entsprechenden DSL (Domain Specific Language) eingesetzt werden kann, um eindeutige und verständliche Anforderungsspezifikationen mit Varianten zu erstellen.

Thomas Stahl ist als Chefarchitekt bei der b+m Informatik AG tätig. Ein Arbeitsschwerpunkt ist die modellgetriebene Software-Entwicklung. Er besitzt langjährige Projekterfahrung auf diesem Gebiet, ist Autor eines entsprechenden Buches sowie Begründer des Open Source Generator Frameworks openArchitectureWare. Weitere Schwerpunkte liegen auf den Themen SOA und BPM.

Tim Weilkiens ist Bereichsleiter Systems-Engineering bei der oose Innovative Informatik GmbH und Autor zahlreicher Buch- und Zeitschriftenpublikationen, darunter das Buch „Systems-Engineering mit SysML/UML“ (dpunkt.verlag, www.system-modellierung.de).

Seine Schwerpunkte liegen im Systems Engineering, der Modellierung und bei Vorgehensmodellen. Er ist zertifizierter V-Modell Projektleiter und QS-Verantwortlicher, aktives Mitglied der Standardisierungs-Arbeitsgruppen zu SysML und UML und hat Teile der SysMLSpezifikation geschrieben.

SysML/DSL für Anforderungen mit Varianten

Motivation



- Variable Systemteile mit SysML beschreiben
- Einfache Sichten auf das Modell bieten
- Trennung der Entwicklungssicht und der Produktsicht



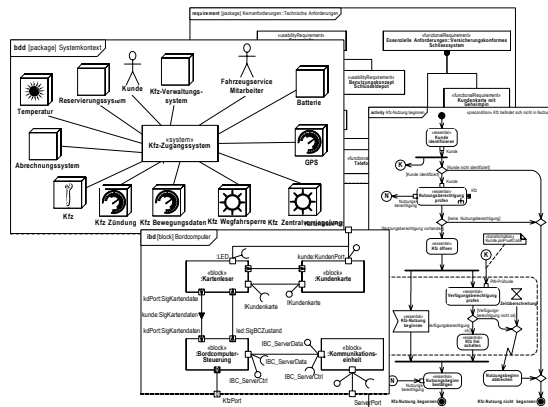
Variabilität in der Produktlinienentwicklung

Variabilität in der Produktentwicklung

SysML/DSL für Anforderungen mit Varianten

Was ist SysML?

OMG Systems Modeling Language (SysML) ist eine Modellierungssprache zur Spezifikation, Analyse, Design und Test von Systemen.



SysML/DSL für Anforderungen mit Varianten

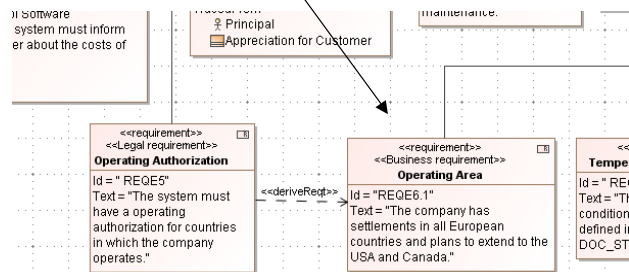
Anforderungen modellieren vs. textuell beschreiben

Modellbasierte Anforderungsmodellierung bedeutet, den Anforderungstext in ein Modell zu integrieren. Die modellierte Anforderung kann das Original sein oder eine Referenz auf eine externe Anforderung, z.B. in DOORS™.

ANFORDERUNG	Kategorie	Text
REQE6.1	Operating Area	The system must have the capability to be used internationally even if it is not connected with the company IT system.
REQE5	Operating Authorization	The company has settlements in all European countries and plans to extend to the USA and Canada.
REQE5	Operating Authorization	The system must have a operating authorization for countries in which the company operates.
REQT1	Power	Power Management of the system.
REQT1.2	Power Saving	The system must switch off all power consumers that are not in

Vorteile:

- Verfolgbarkeit
- Statische Analysen
- Anforderungen stehen dort, wo sie benötigt werden



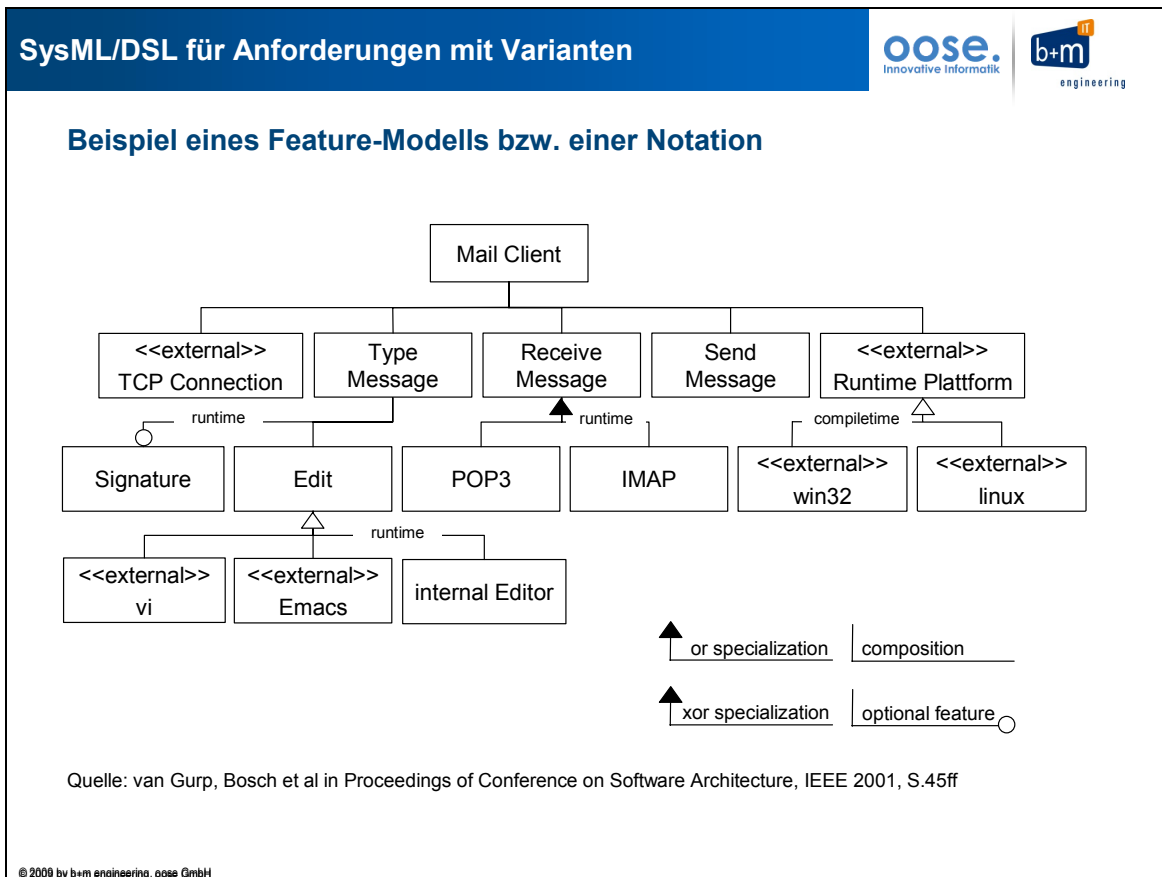
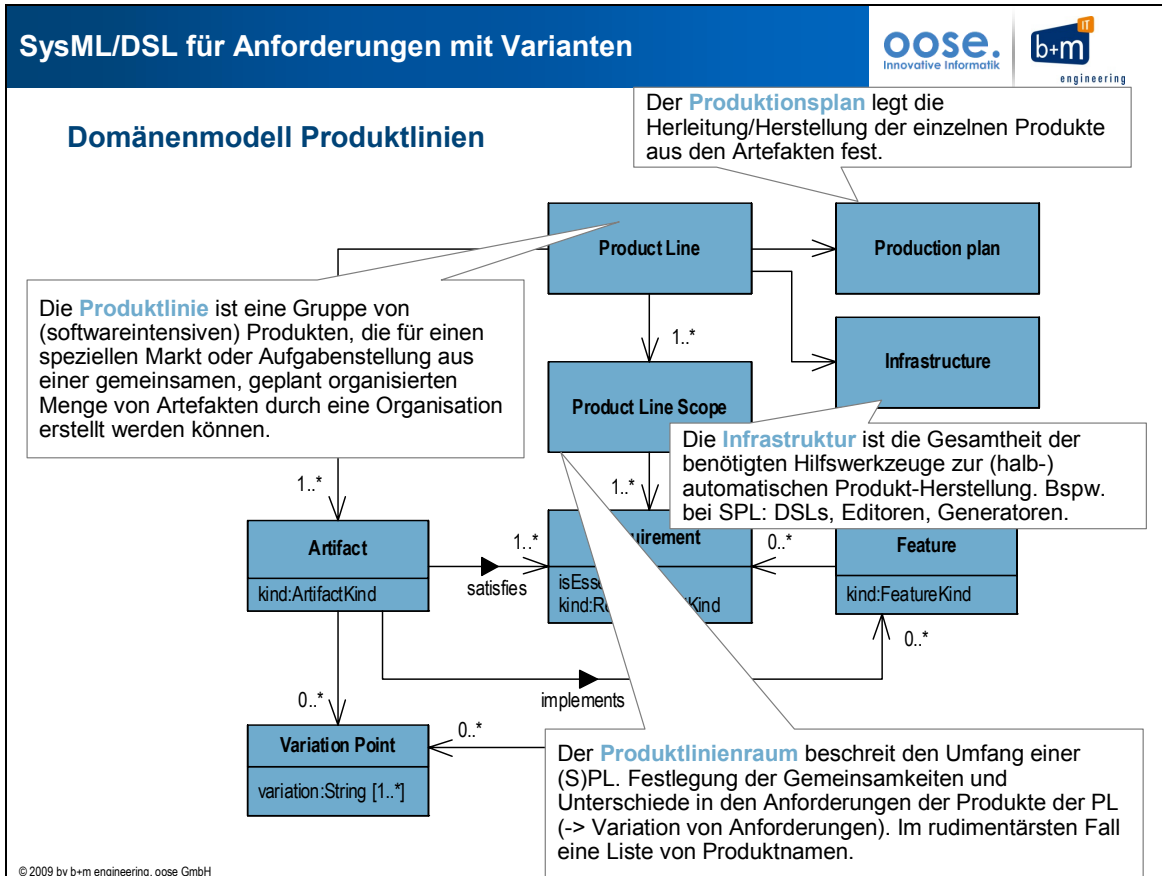
© 2009 by b+m engineering, oose GmbH

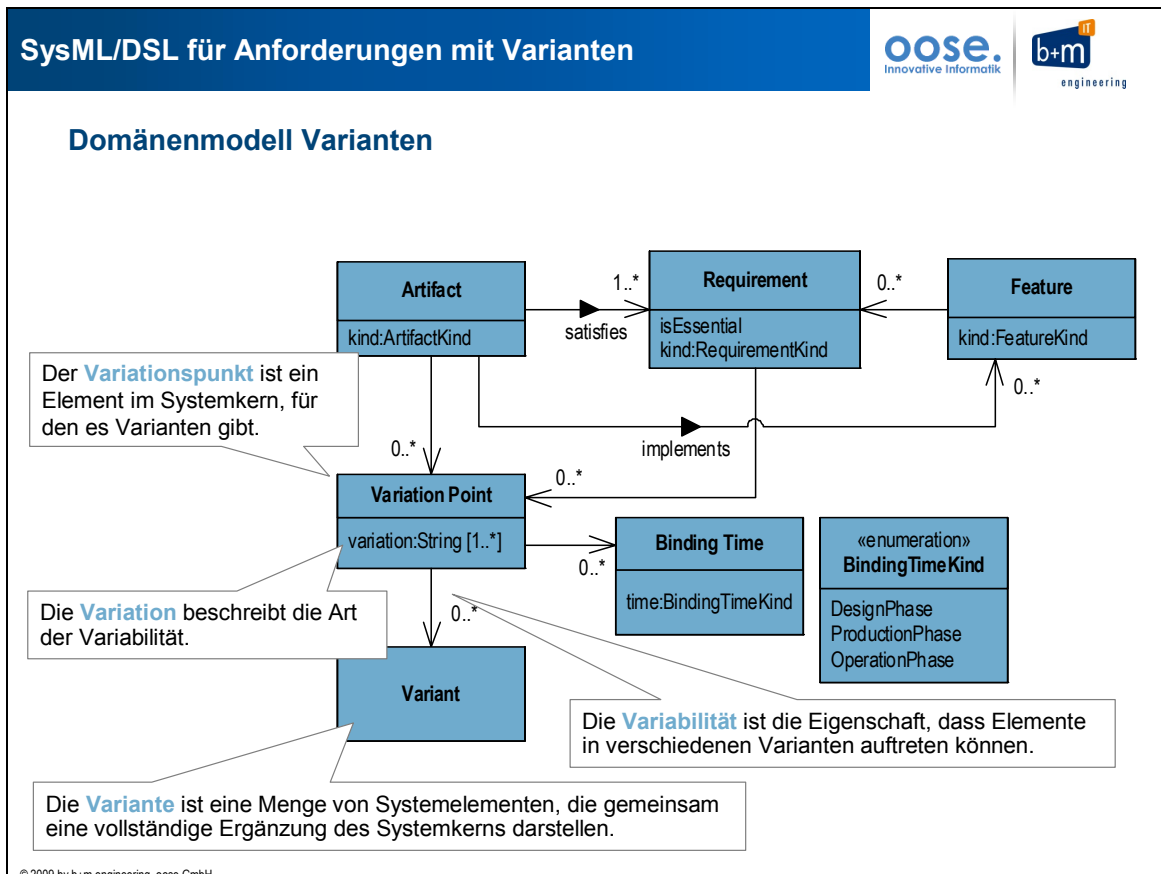
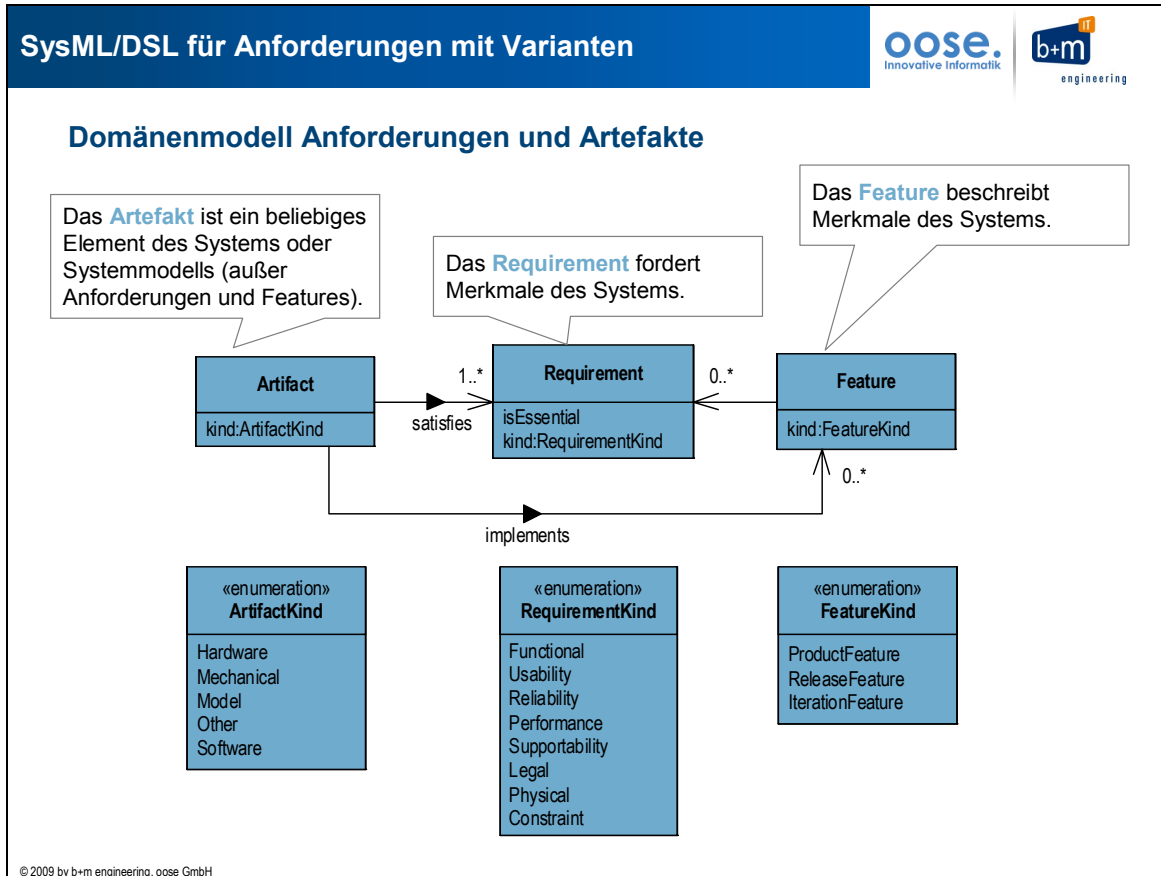
SysML/DSL für Anforderungen mit Varianten

(Software-) Product Line Engineering – Grundlagen

- Treiber: Kostenreduktion, Effizienzsteigerung im Kontext der Herstellung verwandter (Software-)Produkte
- Ansatz: Organisierte Wiederverwendung und ggf. automatisierte Herstellung in einem überschaubaren/definierten Scope (d.h. kein „globaler“ Re-Use Anspruch)
- Schlüsselkonzept: Identifikation von Gemeinsamkeiten und Unterschieden für eine Menge „ähnlicher“ Produkte (-> Produktlinie)
 - Gemeinsamkeiten auf die gleiche Art und Weise und mit gleichen Mitteln herstellen bzw. (bei Software) direkte Wiederverwendung
 - Mögliche Varianten in den Anforderungen erfassen und die Implikationen auf Systemteile und Herstellungsprozess analysieren
- Automation:
 - Herstellung einer „Produktionsstraße“ auf dieser Basis
 - Steuerung der Produktionsstraße durch Endbenutzer-verständliche Spezifikationen (-> Problemraum vs. Lösungsraum)
- Trennung von Product Engineering und Product Line Engineering

© 2009 by b+m engineering, oose GmbH





SysML/DSL für Anforderungen mit Varianten

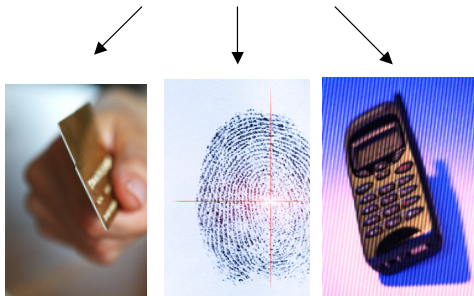


Beispiel Kfz-Zugangssystem für Vermietungsunternehmen



Variation Kundenidentifikation

Variation Komfortfunktionen

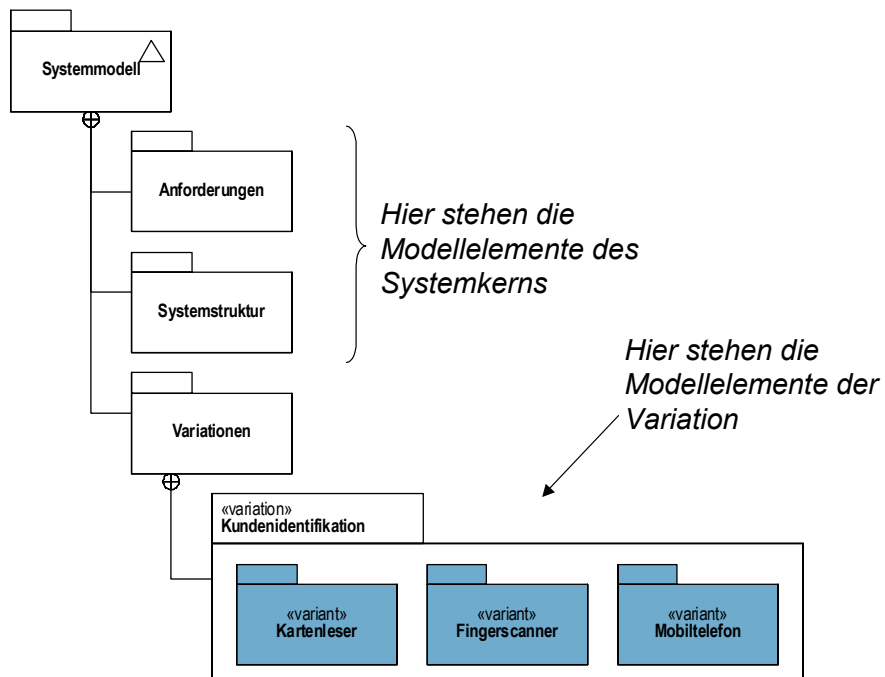


© 2009 by b+m engineering, oose GmbH

SysML/DSL für Anforderungen mit Varianten



Beispiel: Variationen und Varianten

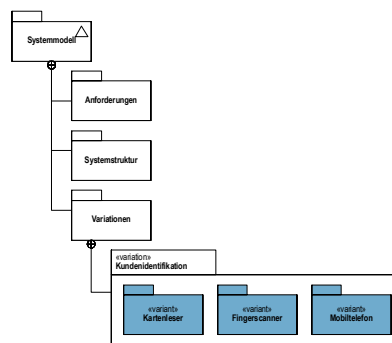


© 2009 by b+m engineering, oose GmbH

SysML/DSL für Anforderungen mit Varianten

Separation of Concerns: Kern und Varianten

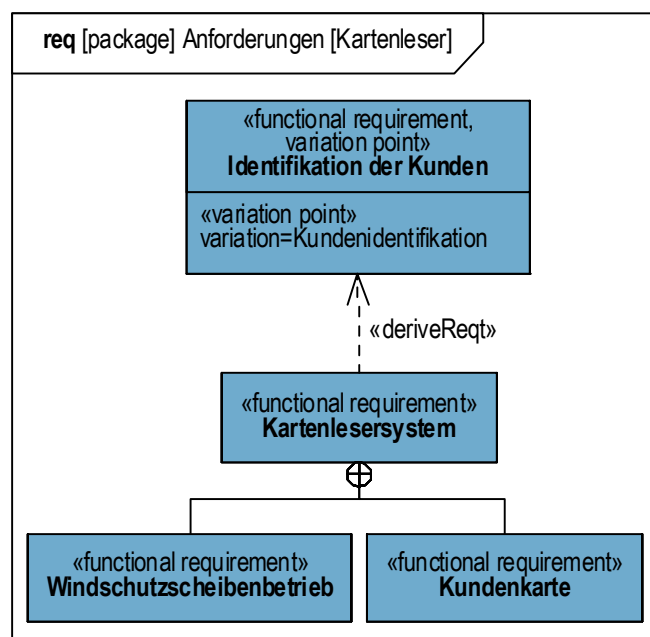
- Getrennte statische Analysen des Kerns und der Varianten sind möglich
 - Beispielsweise durch Anreicherung des Modells mit Leistungsmerkmalen mit Hilfe eigener Stereotypen oder des MARTE-Profiles
 - MOE-Funktionen (*Measures of Effectiveness*) zur Entscheidungsfindung der besten Variantenkonfiguration
- Deutliche und bewusste Trennung der allgemeinen und variablen Systemteile.
- Kompliziert ist es nur an der Trennlinie Kern/Varianten.



© 2009 by b+m engineering, oose GmbH

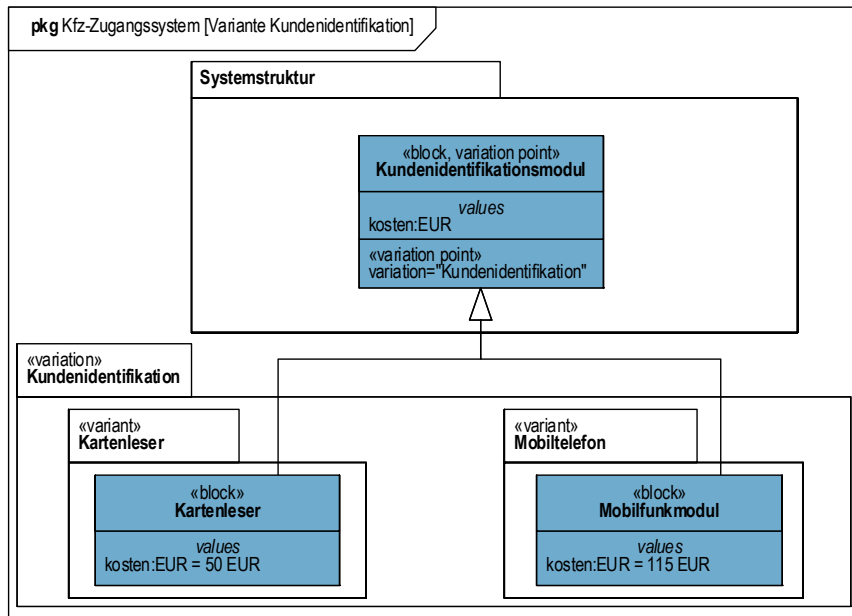
SysML/DSL für Anforderungen mit Varianten

Beispiel: Variationspunkt an einer Anforderung



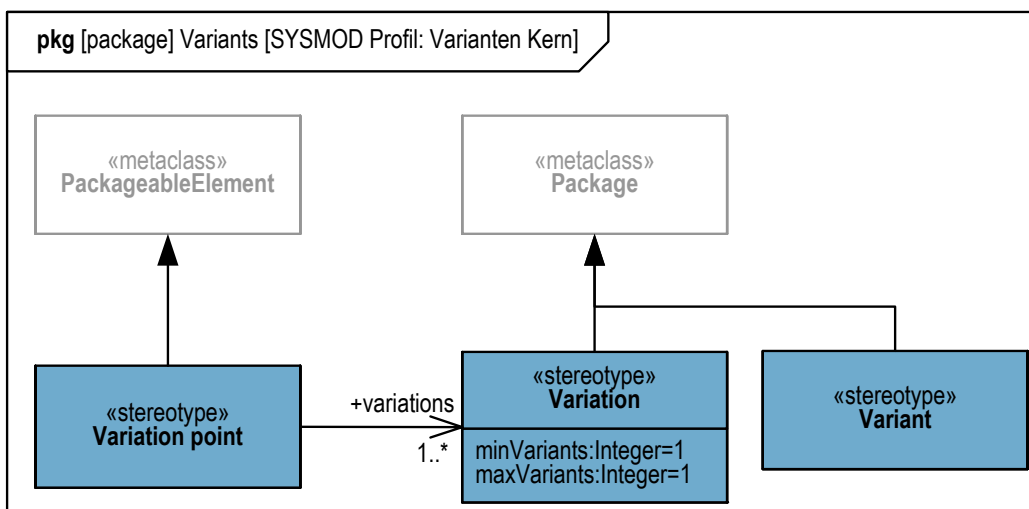
© 2009 by b+m engineering, oose GmbH

Beispiel: Variantenmodellierung im Design





© 2009 by b+m engineering, oose GmbH

Variantenprofil für SysML



© 2009 by b+m engineering, oose GmbH

SysML/DSL für Anforderungen mit Varianten



Beziehungen zwischen Varianten- und Systemkernelementen

Systemkern	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 5px; width: 20%;">«functional requirement, variation point» Identifikation der Kunden</div> <div style="border: 1px solid black; border-radius: 50%; padding: 5px; width: 20%;">«variation point» Kfz-Nutzung beginnen</div> <div style="border: 1px solid black; padding: 5px; width: 20%;">«activity, variation point» Kunde identifizieren</div> <div style="border: 1px solid black; padding: 5px; width: 20%;">«block, variation point» Kundenidentifikator</div> </div>
«variant» Kartenleser	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 5px; width: 20%;">«functional requirement» Kartenlesersystem</div> <div style="border: 1px solid black; border-radius: 50%; padding: 5px; width: 20%;">Kunde mit Karte identifizieren</div> <div style="border: 1px solid black; padding: 5px; width: 20%;">«activity» Kunde mit Karte identifizieren</div> <div style="border: 1px solid black; padding: 5px; width: 20%;">«block» Kartenleser</div> </div>

- Keine festen Vorgaben der Beziehungsarten
- Freiheitsgrade bzgl. Beziehungsart hängen von der geplanten Verwendung des Modells ab
 - Automatisierte Analysen
 - Modelltransformationen
 - Nur für Menschen zum Lesen

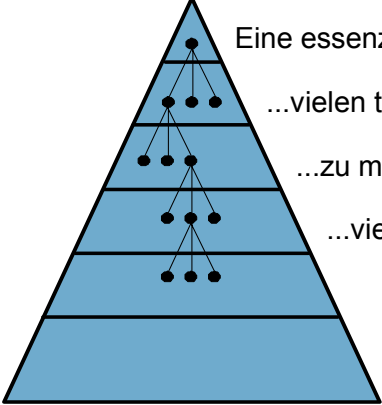
© 2009 by b+m engineering, oose GmbH

SysML/DSL für Anforderungen mit Varianten

Essenzielle vs. Technische Anforderungen

Die **essenzielle Anforderungen** beschreiben die rein fachliche Intention und sind unabhängig von der technischen Umsetzung (Lösung).
 Entsprechend reden wir von einer **technischen Anforderung**, wenn sie auf einem Lösungsansatz basiert.



Eine essenzielle Anforderung führt zu...

...vielen technischen Anforderungen, die jeweils...

...zu mehreren Analyseelementen und...

...vielen Designelementen führen.

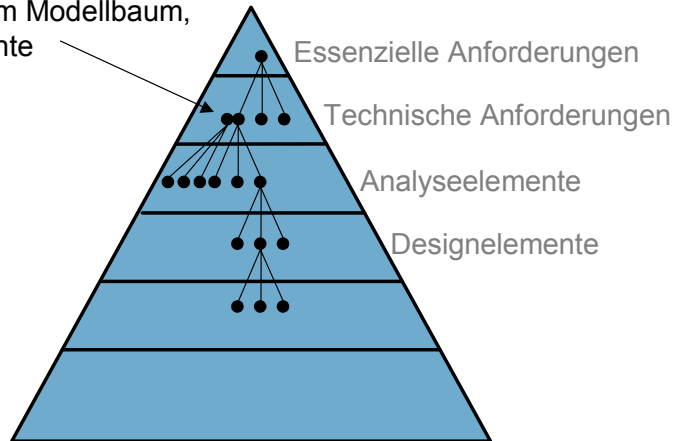
© 2009 by b+m engineering, oose GmbH

SysML/DSL für Anforderungen mit Varianten



Vervielfachung durch Variabilität

Je höher die Variabilität im Modellbaum, desto mehr Modellelemente

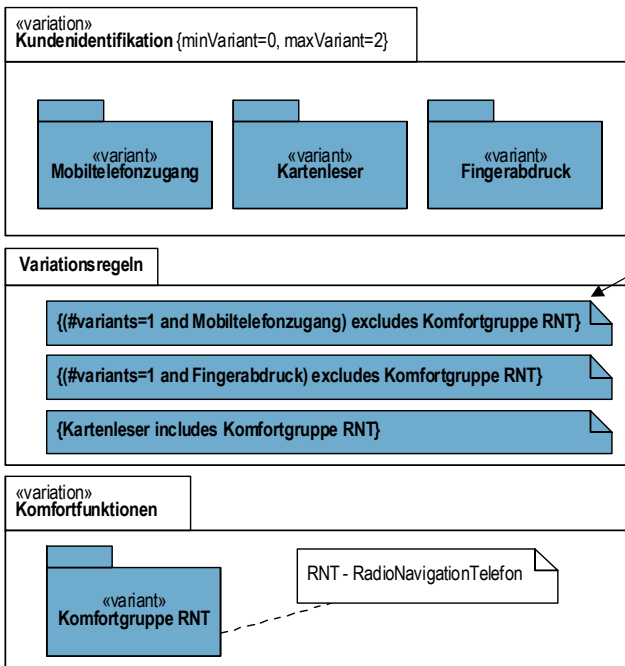


Ein Systemmodell ist häufig komplex. Das Modell eines Konfigurationsraums ist komplexer.

SysML/DSL für Anforderungen mit Varianten



Beispiel: Variationen konfigurieren

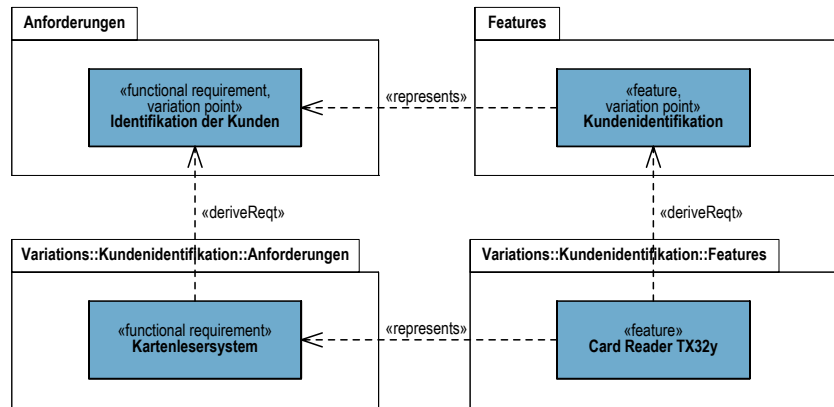
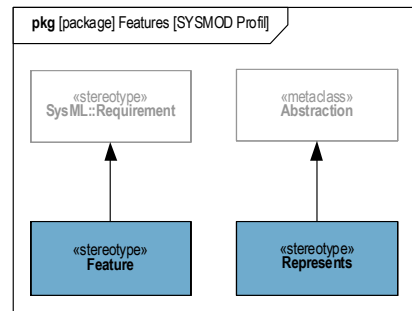


Regeln formuliert in einer textuellen DSL und eingebettet in UML-Zusicherungen definieren die Variationsmöglichkeiten

SysML/DSL für Anforderungen mit Varianten

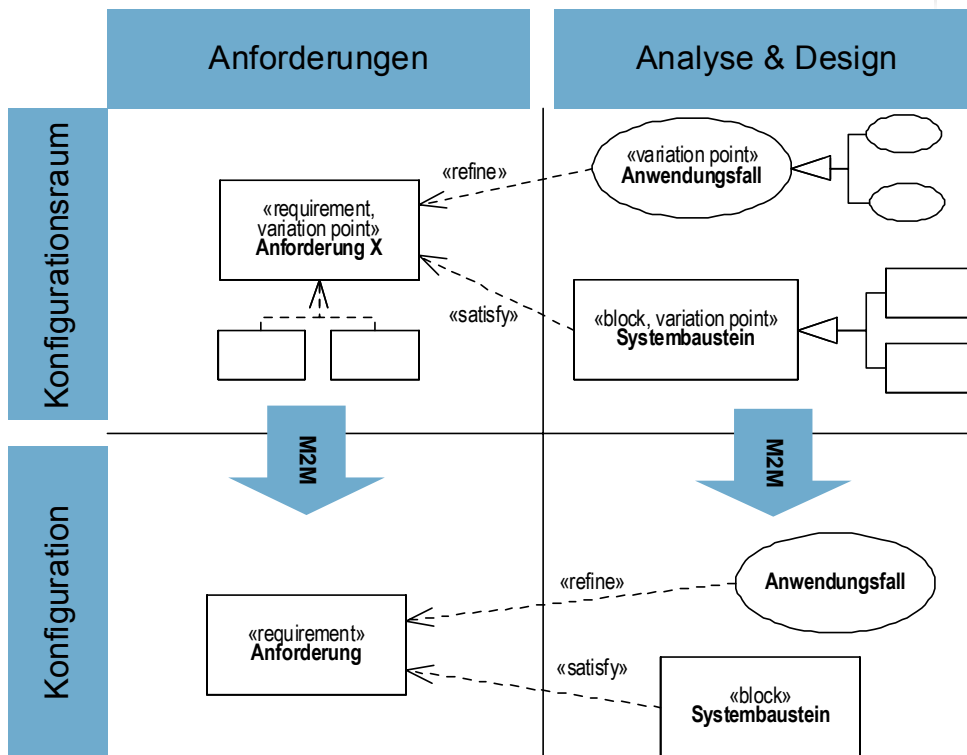
Featureprofil mit Beispiel

- Variabilität über Variantenstereotypen
- Brückenschlag zum Feature- und Projektmanagement



© 2009 by b+m engineering, oose GmbH

SysML/DSL für Anforderungen mit Varianten



© 2009 by b+m engineering, oose GmbH

Model2Model-Transformation (M2M)

Bereits einfache Variationen führen zu komplexen Konfigurationsräumen. Erforderlich ist eine einfache Sicht für eine ausgewählte Konfiguration. Die Sicht kann von einer M2M-Transformation erstellt werden.

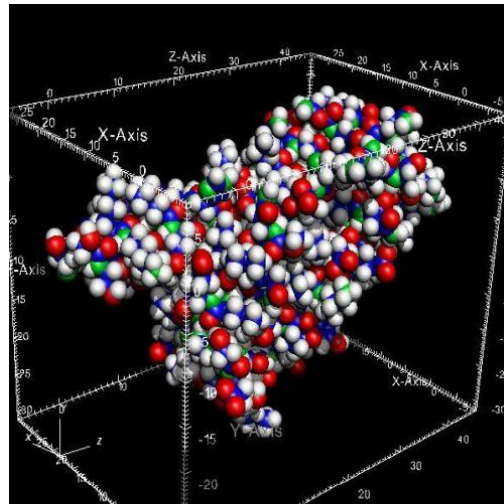
3 Variationen führen zu einem 3-dimensionalen Konfigurationsraum und ggf. zu vielen möglichen Konfigurationen.

Eine Konfiguration ist ein Punkt im Konfigurationsraum.

Ziel der Modell-Transformationen:
Ausblendung irrelevanter Details

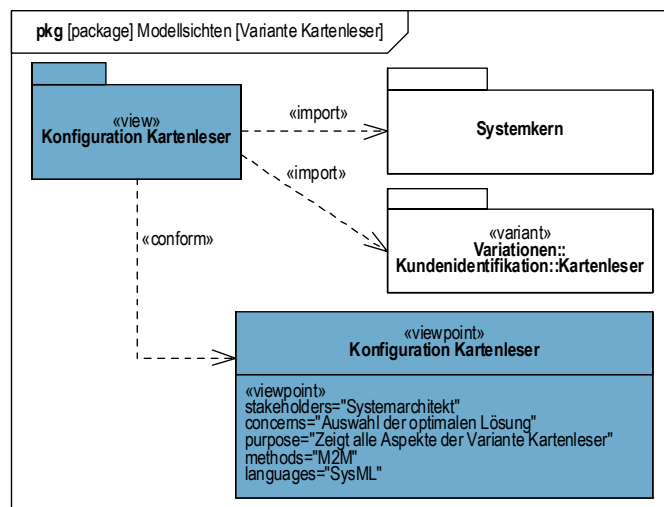


- Erzeugung eines Produktmodells aus dem Produktlinien-Modell
- Eliminierung zukünftig nicht mehr existenter Varianten und ggf. Auflösung dadurch überflüssig gewordener Abstraktionen



Modellsicht und Standpunkt (View/Viewpoint)

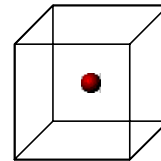
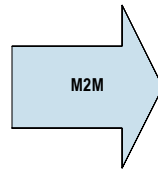
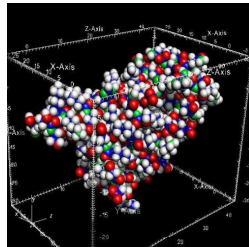
Der Standpunkt definiert die Modellsicht. Es werden die Stakeholder, ihre Ziele und Regeln zur Erstellung der Modellsicht berücksichtigt.



SysML/DSL für Anforderungen mit Varianten

Kategorisierung der Model2Model-Transformationen

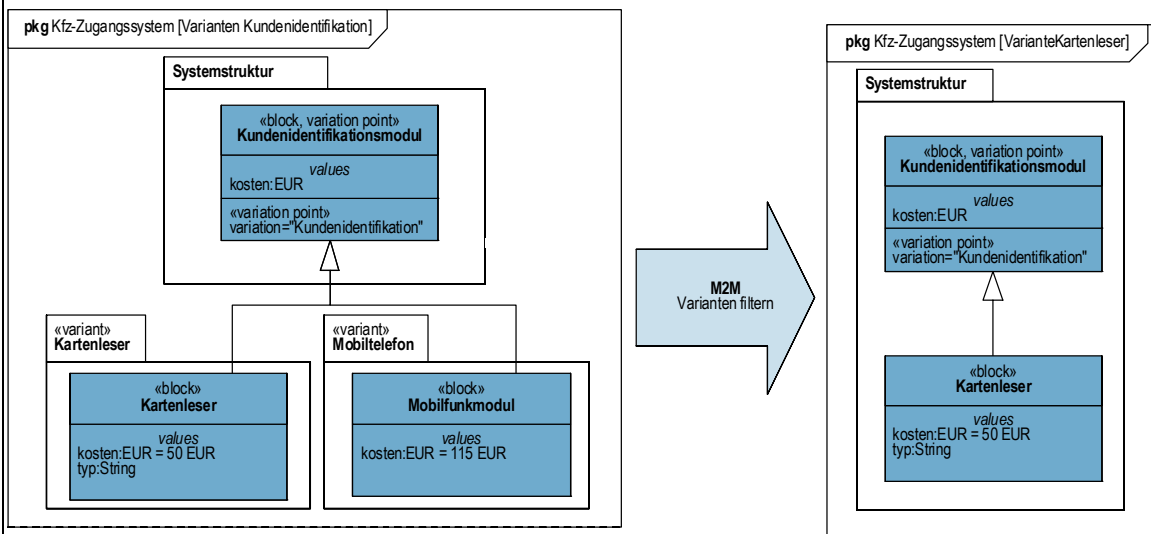
- View- vs. Copy-Strategie:
 - View: Die Transformation erzeugt eine View im Ausgangsmodell. Elemente, die nur referenziert werden, können nicht verändert werden.
-> Trennung von Product Line Engineering und Product Engineering
 - Copy: Die Transformation erzeugt ein neues Modell und sorgt für Traceability
-> Verwerfen von Varianten während der Entwicklung
- Filter vs. Refactoring-Strategie
 - Filter: Nicht mehr benötigte Modellelemente werden durch die Transformation (aus der View oder der Kopie) entfernt
-> einfache Handhabung/Umsetzung aber entfernt ggf. nicht allen „Ballast“
 - Model-Refactoring: Es gibt nicht „die eine“ Transformation, sondern einen Satz adäquater Refactorings, mit jeweils einer, ggf. nicht-trivialen Transformation
-> bestmögliche Reduktion, u.U. aber komplex in der Umsetzung



© 2009 by b+m engineering, oose GmbH

SysML/DSL für Anforderungen mit Varianten

Beispiel: Einfacher M2M-Ansatz

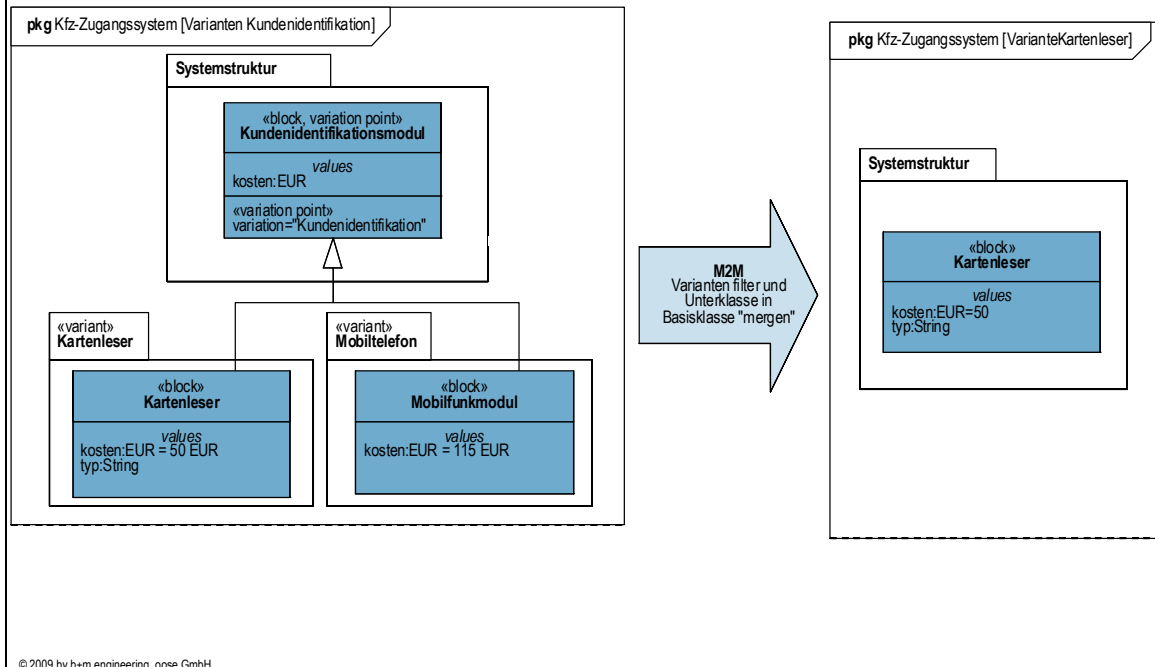


© 2009 by b+m engineering, oose GmbH

SysML/DSL für Anforderungen mit Varianten

oose.
Innovative Informatikb+m
engineering

Beispiel: M2M-Model-Refactoring-Ansatz



(Be-)Merkenwertes zum Abschluss

- Varianten sind notwendig für Produktlinien und für Designentscheidungen.
- Der Systemkern ist unabhängig von den Varianten.
- Es gilt: Systemkern + Varianten = Produkt.
- Modellanalysewerkzeuge können Varianten untereinander vergleichen beispielsweise hinsichtlich Kosten, Energieverbrauch, Leistung.
- M2M-Transformationen können einfache Sichten auf Konfigurationen erzeugen.
- Die Modellierung von Varianten hat hohe Ansprüche an Werkzeug und Mensch.



© 2009 by b+m engineering, oose GmbH

Vielen Dank für Ihre
Aufmerksamkeit!
REConf 2009, München

Tim Weilkiens

Bereichsleiter
Systems-Engineering

tim.weilkiens@oose.de



Thomas Stahl

Chefarchitekt

t.stahl@bmiag.de

