

# Wie, Weshalb und Wann?

## Eine Diskussion

Danilo Beuche, pure-systems GmbH  
Enrico Schnepel, b+m Informatik AG

## Wer sind wir?

### Danilo Beuche

- PLE
- pure-systems GmbH
- <http://www.pure-systems.com/>

### Enrico Schnepel

- MDD und DSL
- b+m Informatik AG
- <http://engineering.bmiag.de/>



## Was denken wir?

### PLE+MDD = gute Idee

- Gute Kombination des erhöhten Abstraktionslevels
- Erfassung der Produkteigenschaften durch PLE in einem Modell
- MDD ermöglicht einfache Partitionierung und Zusammenfassung von Variabilität

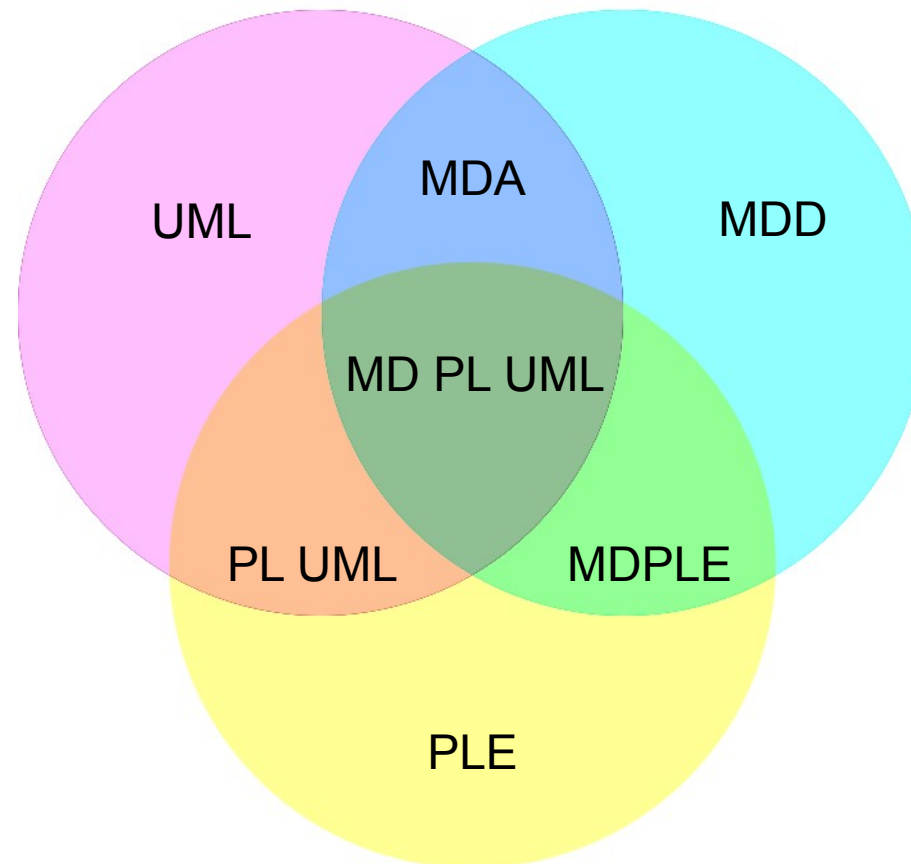
### PLE+UML = {gute|schlechte}? Idee

- Was ist der Mehrwert der Nutzung von UML-Modellen?
- UML am Anfang und/oder am Ende des Entwicklungsprozesses?
- Und was ist mit UML eigentlich gemeint?

## Wofür wird UML verwendet?

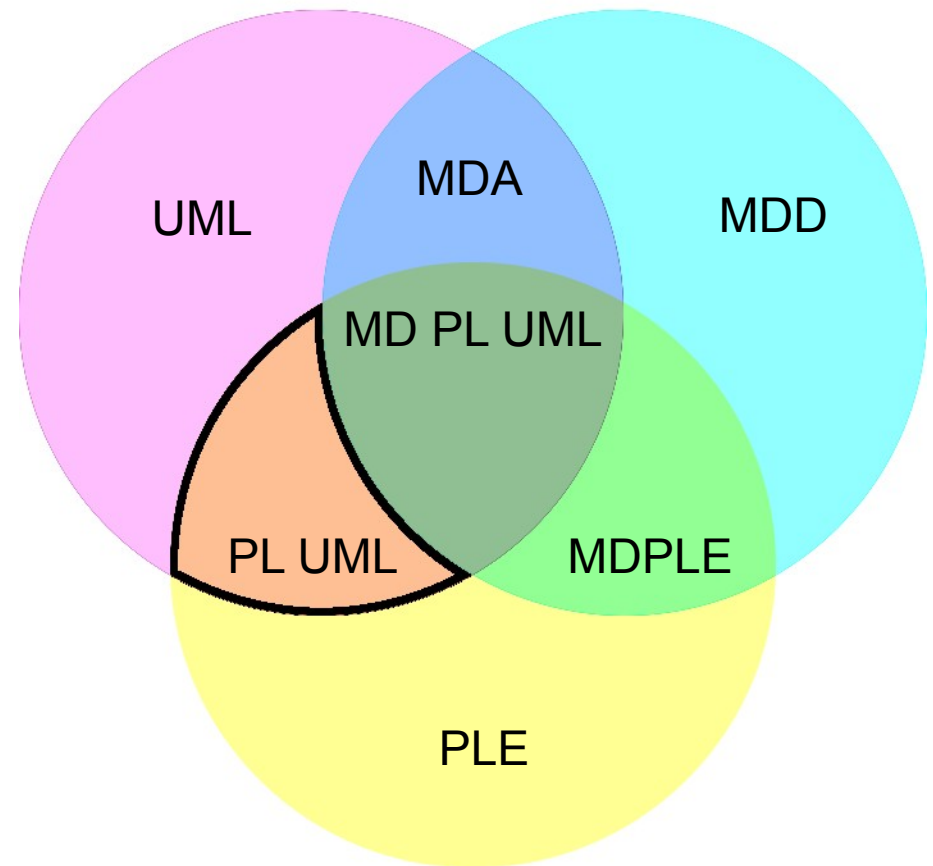
- Kommunikation:
  - Beschreibung Architektur (Klass. UML)
  - Beschreibung von Anforderung (SysML)
- Systemgenerierung:
  - Erzeugung von Code-Fragmenten (Klassendiagramme)
  - Erzeugung von Anwendungscode ( MDA )
  - Prototyping/Simulation (Executable UML)

# Wie stehen UML, MDD und PLE im Verhältnis?



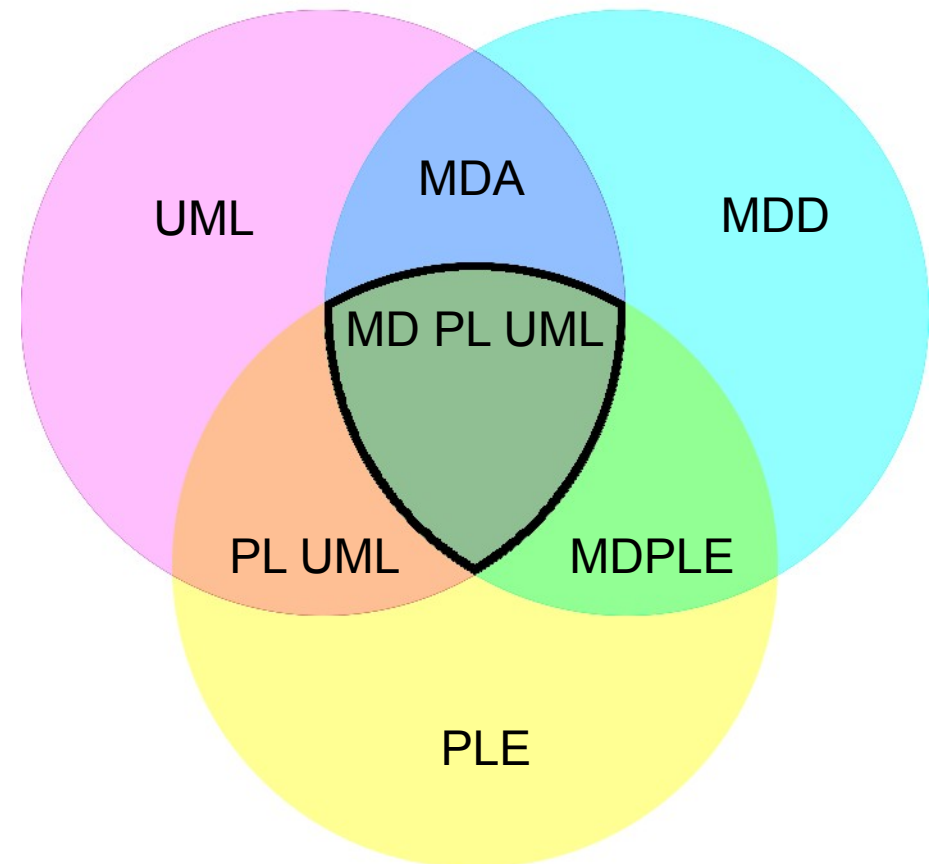
## Produktlinien mit UML

- Nutzung von (Standard) UML Modellen als Grundlage für PLE
- Keine automatische Erstellung essentieller Artefakte der Implementierung
- Beispiele:
  - Conditional UML
  - Nutzung von speziellen Variabilitätsprofilen (Gooma, Stahl/Wilkiens)



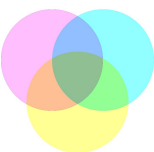
# Model Driven Produktlinien mit UML

- Nutzung von UML Modellen als Grundlage für PLE
- Automatische Erstellung essentieller Artefakte der Implementierung
- Beispiele:
  - Nutzung von domänenbezogenen Profilen mit integriertem Variabilitätskonzept
  - Nutzung von domänenbezogenen Profilen mit Conditional UML



# Nutzungsszenarien – Matrix

|                                 | UML | MDD | PLE | MDA | MDPLE | PLUML | MDPLUML |
|---------------------------------|-----|-----|-----|-----|-------|-------|---------|
| Dokumentation Architektur       | ■   | ■   | ■   | ■   |       | ■     | ?       |
| Anforderungen System            | ■   | ■   | ■   |     | ■     | ■     | ?       |
| Generierung von Code-Fragmenten | ■   |     | ■   |     |       | ■     | ?       |
| Umfassende Generierung von Code |     | ■   |     | ■   | ■     | ■     | ?       |



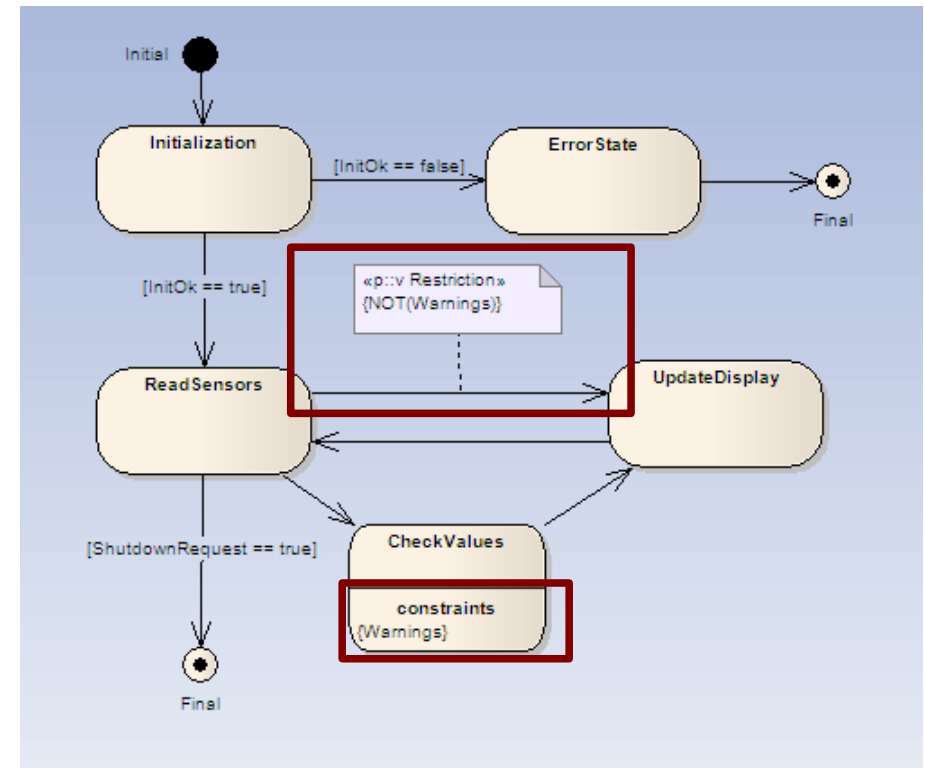
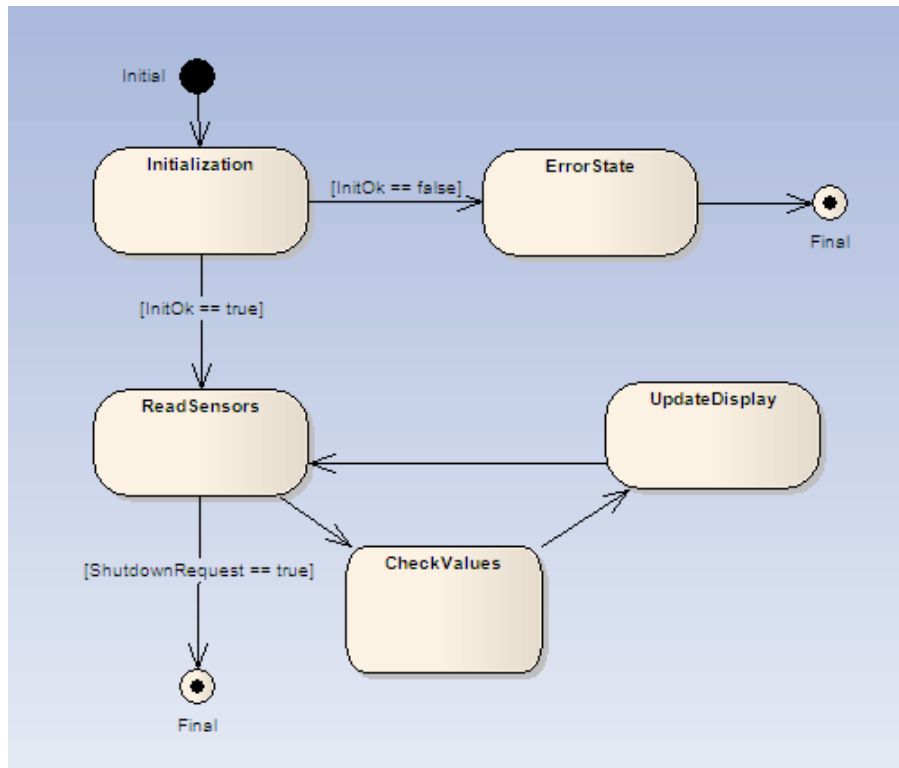


## Umsetzung PL UML: Conditional UML

- Übertragung der Idee von „`#ifdef`“ in die UML Welt:
  - UML Constraints steuern Vorhandensein von UML Artefakten in den Modellinstanzen der Varianten
  - Constraints referenzieren Feature und andere Konfigurationsinformationen
  - Featuremodelle können separat oder in UML modelliert werden



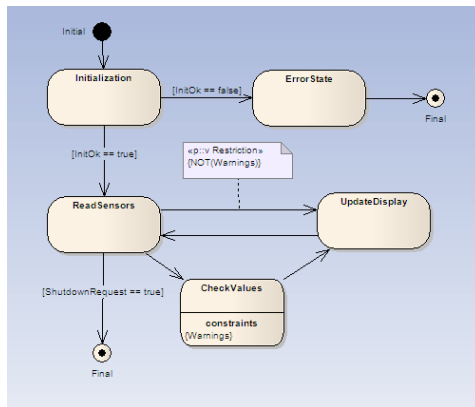
# Conditional UML: 1. Schritt



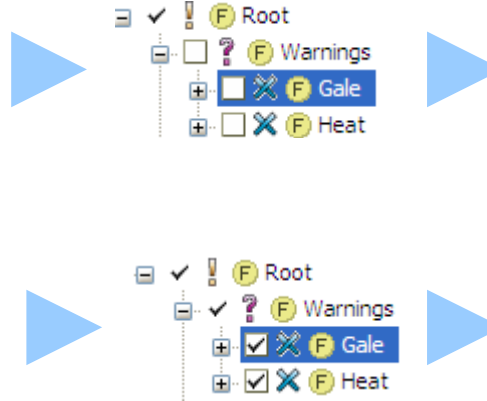
Hinzufügen von Variabilität durch optionale UML Elemente und spezielle UML Constraints



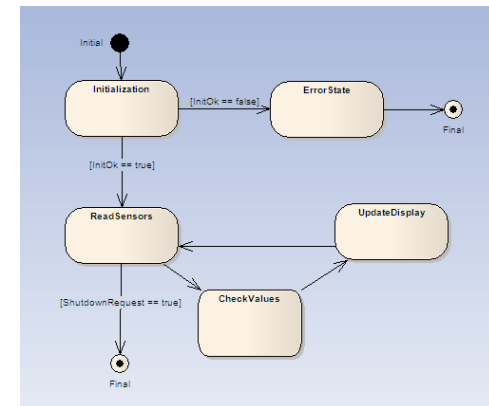
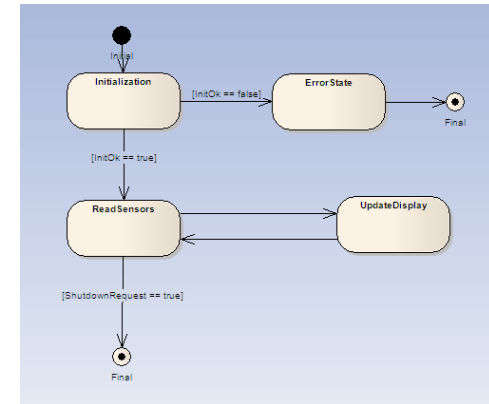
# Conditional UML: 2. Schritt



PL UML Modell(e)



Variantenbeschreibung



Auflösen von Variabilität durch Instanzierung der UML Modelle anhand von Variantenbeschreibungen



## Umsetzung PL UML: Conditional UML

- Pro:
  - Kann grundsätzlich mit allen UML Modellen angewandt werden
  - Einfaches Konzept
  - Varianten UML Modelle sind „Standard“ UML Modelle
- Con:
  - nur „negative“ Variabilität (Weglassen von Optionen) modellierbar
  - Viele existierende semantische Checks nur auf den Instanzen (Varianten UML Modellen) anwendbar



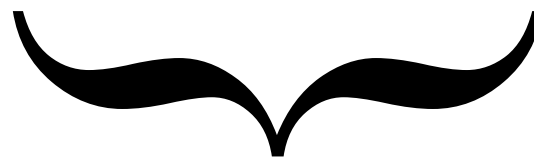
## PLE und MDD

### MDD

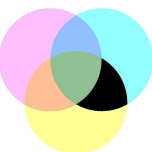
- Nutzung der PLE-Modelle
- Konfiguration durch Varianten
- Evtl. vorhandene PLE-Tools

### PLE

- Einfache Generierung von SW-Artefakten mit der MDD
- Evtl. Modelltransformation für existierende Generatoren

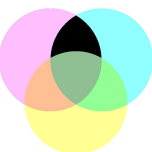


Vorteile auf beiden Seiten



## Ist UML MDD?

- Profiliertes UML
- MDA (PIM / PSM / Generierung)
- Essentieller Bestandteil des Entwicklungsprozesses
- Generierung wichtiger Teil- oder ganzer Artefakte
  
- ✓ DSL-Design Bestandteil von UML-Tools
- x Keine M2M-Transformationen
- x UML-Tool basierter Code-Generator



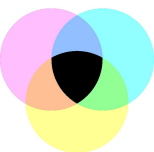
## UML und PLE?

- ✓ Einführung von Modellen (DSL mit Profilen)
- ✓ Generierte finale Dokumentation
  
- x Sprachergonomie
  - Lesbarkeit / Verständlichkeit
  - UML ≠ DSL
- x Abhängigkeit vom UML-Tool
  - PLE → lange Nutzungsdauer
- x Ursprüngliche Intention



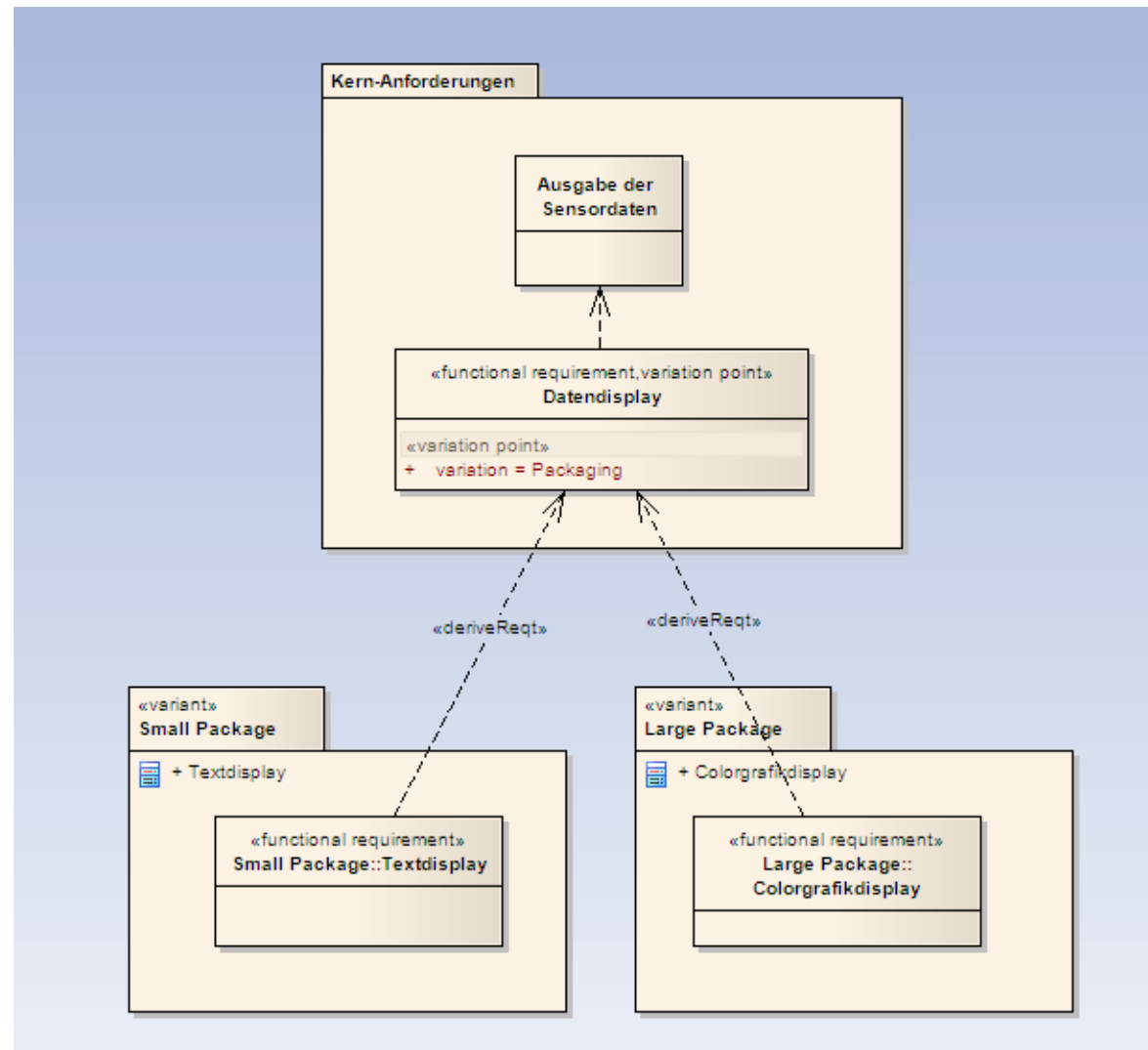
## Umsetzung MD PL UML: Profiled SysML

- Nutzung eines Profils zur Darstellung von Variabilität
  - Stereotypen und zugehörige Semantik ergänzen UML um Variabilitätsbeschreibungsmöglichkeit
  - Beispiel nutzt Profil definiert von Stahl/Weilkiens
    - Markieren von <<variation points>> in Modellen und verbinden mit den Instanzen des Variationspunktes
    - Informationen zu Instanzen von Variationspunkten werden in <<variant>> Packages in SysML abgelegt





# Profiled SysML: Beispieldiagramm



## Abschlußbetrachtungen

- Für wen ist UML und PLE eine sinnvolle Kombination?
  - „Legacy UML“, insbesondere wenn zur Generierung verwendet
  - Werkzeugkenntnisse/-verfügbarkeit in der Organisation
- Ist MD PL UML eine gute Idee? Oder besser gleich „richtige“ PL-DSL
  - wird bereits UML verwendet: siehe oben
  - bei neuen Projekten: ???

## Wo gibt es mehr?

- Lesen
  - Hassan Gomaa: Designing Software Product Lines with UML, ISBN-10: 0201775956
  - Stahl/Weilkiens: SysML/DSL für Anforderungen mit Varianten  
[http://2009.reconf.de/fileadmin/PDF\\_Dateien/REConf\\_2009/Vortraege/Weilkiens\\_Stahl.pdf](http://2009.reconf.de/fileadmin/PDF_Dateien/REConf_2009/Vortraege/Weilkiens_Stahl.pdf)
  - F. Puhmann et al: Variability Mechanisms for Process Models.  
[http://www.pesoa.de/pages/Publications/Fachberichte062005/PESOA\\_TR\\_17-2005.pdf](http://www.pesoa.de/pages/Publications/Fachberichte062005/PESOA_TR_17-2005.pdf)
- Ausprobieren:
  - pure::variants for Enterprise Architect (ab 1.6.2009)  
<http://www.pure-systems.com/>