

# Model Driven Software Development

## MDSD Einführung und Überblick

**Referent:**  
**Carsten Schädel**

# Ziele

- **Grundgedanke**
- **Glossar der wichtigsten Begriffe**

# Glossar

- **Domäne**

# Domäne

## ● mögliche Definitionen:

- „A domain is a bounded area of knowledge or interest“  
(nach Völter)
  
- „abgrenzbares Problemfeld des täglichen Lebens in  
Softwaresystemen“  
(Wiktionary)

## ● je abgegrenzter, desto besser bzw. einfacher hand zu haben.

# Glossar

## ● Domäne

➔ abgegrenzter Problemraum

# Problemfelder/ Problemräume sind ...

**architektur-zentriert** ←  
(Architektur/Technik motiviert)

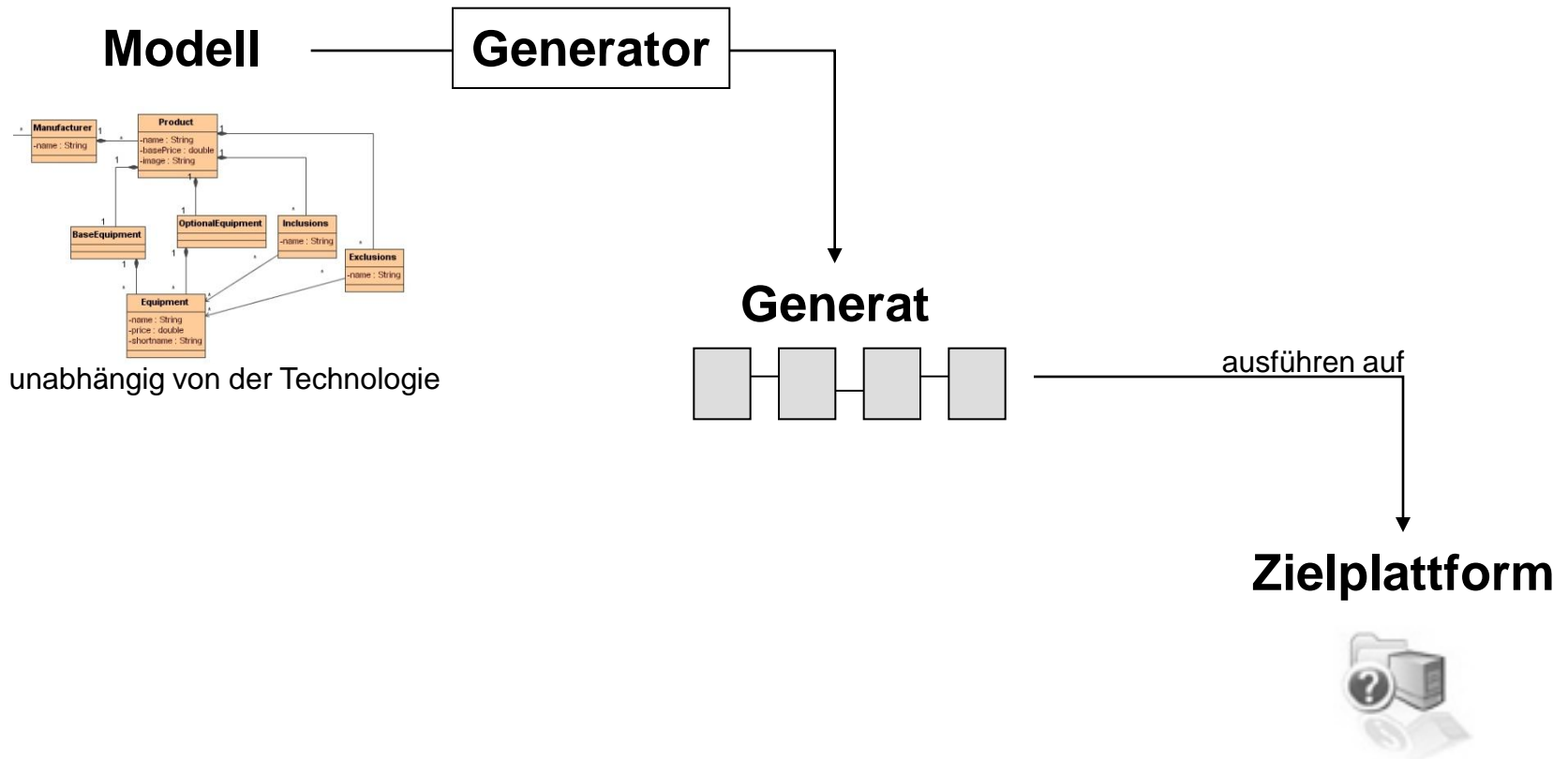
- Abbildung in die gewählte Technologie
- architektur-zentrierte MDSD

→ **domänen-zentriert**  
(fachlich motiviert)

- wird durch die konkrete Fachlichkeit vorgegeben
- fachlich-zentrierte MDSD

# Grundgedanke

# Grundgedanke





# Glossar

## ● Domäne

→ abgegrenzter Problemraum

# Glossar

## ● Domäne

→ abgegrenzter Problemraum

## ● Modell

- formalisiertes Wissen der Domäne bzw. fachliche Beschreibung eines Problemraumes
- unabhängig von der einzusetzenden Technologie

# Glossar

- **Domäne** → abgegrenzter Problemraum
- **Modell** → formalisiertes Wissen der Domäne
- **Generator**
- **Transformator**

# Generator - Transformator

## ● Gemeinsamkeiten

- interpretieren eines Modells
- enthalten möglichst viel Wissen um den Problemraum
- setzen ein Regelwerk zur Generierung/Transformation um
- setzen allgemeine Muster für Architektur und Code um

# Generator - Transformator

- **Generatoren M2T (model to text)**

- beschreiben Generierung von Artefakten  
(HTML-Seiten, XML-Dateien, Quellcode ...)

- **Transformatoren M2M (model to model)**

- beschreiben Modell-zu-Modell-Transformationen
- tiefgehende Informationen: Artikel im Java - Magazin

- Generatoren/Transformatoren können in beliebig vielen Schritten gekoppelt sein

# Glossar

- Domäne → abgegrenzter Problemraum
- Modell → formalisiertes Wissen der Domäne
- Generator (M2T) → Generierung von Artefakten
- Transformator (M2M) → Modell – zu – Modell

# Glossar

- **Domäne** → abgegrenzter Problemraum
- **Modell** → formalisiertes Wissen der Domäne
- **Generator (M2T)** → Generierung von Artefakten
- **Transformator (M2M)** → Modell – zu – Modell
- **Generat**
  - Ergebnis der Generierung
  - wird auf Zielplattform ausgeführt

# Glossar

- Domäne → abgegrenzter Problemraum
- Modell → formalisiertes Wissen der Domäne
- Generator (M2T) → Generierung von Artefakten
- Transformator (M2M) → Modell – zu – Modell
- Generat → Ergebnis der Generierung
- Zielplattform



# Zielpattform

- Laufzeitumgebung zur Ausführung der Generate
- wird durch Frameworks und Komponenten zur „domänenspezifischen Plattform“

# Glossar

- Domäne → abgegrenzter Problemraum
- Modell → formalisiertes Wissen der Domäne
- Generator (M2T) → Generierung von Artefakten
- Transformator (M2M) → Modell – zu – Modell
- Generat → Ergebnis der Generierung
- Zielplattform → Laufzeitumgebung
- domänenspezifische Plattform

# Domänenspezifische Plattform

- bringt die Zielplattform durch Frameworks und Komponenten der Domäne ‚näher‘
- je spezifischer die Plattform, desto einfacher der Generator
- für alle Anwendungen einer Domäne gleich

# Glossar

- Domäne → abgegrenzter Problemraum
- Modell → formalisiertes Wissen der Domäne
- Generator (M2T) → Generierung von Artefakten
- Transformator (M2M) → Modell – zu – Modell
- Generat → Ergebnis der Generierung
- Zielplattform → Laufzeitumgebung
- **domänenspezifische Plattform** → **erweiterte Zielplattform**

# Glossar

- **Domäne** → abgegrenzter Problemraum
- **Modell** → formalisiertes Wissen der Domäne
- **Generator (M2T)** → Generierung von Artefakten
- **Transformator (M2M)** → Modell – zu – Modell
- **Generat** → Ergebnis der Generierung
- **Zielplattform** → Laufzeitumgebung
- **domänenspezifische Plattform** → erweiterte Zielplattform
- **DSL**

# Domänenspezifische Sprache

- Sprache zur Formulierung von Modellen in einem Problemraum
  - **DSL** (domain specific language)
- besitzt eine formale Spezifikation in Form eines Metamodells
- kann in unterschiedlichen Formen dargestellt/realisiert werden
  - UML, EMF, textuell
  - konkrete Syntax
- beschreibt **einen** speziellen Problemraum

## DSL in MDSD

- Mehrwert von MDSD steigt,
  - je abgegrenzter der Problemraum und
  - je spezieller die DSL dieses Problemraumes ist
- ▶ Finde abgegrenzten Problemraum und entsprechende DSL
- ▶ In Projekten existieren u.U. **mehrere** DSLs nebeneinander

# Glossar

- Domäne → abgegrenzter Problemraum
- Modell → formalisiertes Wissen der Domäne
- Generator (M2T) → Generierung von Artefakten
- Transformator (M2M) → Modell – zu – Modell
- Generat → Ergebnis der Generierung
- Zielplattform → Laufzeitumgebung
- domänenspezifische Plattform → erweiterte Zielplattform
- DSL → Modellierungssprache im Problemraum
- Metamodell



# Metamodell

- **abstrakte Syntax**
- formalisiert die Strukturen einer Domäne
- Beschreibung der Sprachkonstrukte einer DSL
- definiert die Verwendung der Sprachkonstrukte
- Modelle werden durch ein Metamodell maschinell überprüfbar

# Glossar

- Domäne → abgegrenzter Problemraum
- Modell → formalisiertes Wissen der Domäne
- Generator (M2T) → Generierung von Artefakten
- Transformator (M2M) → Modell – zu – Modell
- Generat → Ergebnis der Generierung
- Zielplattform → Laufzeitumgebung
- domänenspezifische Plattform → erweiterte Zielplattform
- DSL → Modellierungssprache im Problemraum
- **Metamodell** → **Sprachkonstrukte einer DSL**

# Glossar

- Domäne → abgegrenzter Problemraum
- Modell → formalisiertes Wissen der Domäne
- Generator (M2T) → Generierung von Artefakten
- Transformator (M2M) → Modell – zu – Modell
- Generat → Ergebnis der Generierung
- Zielplattform → Laufzeitumgebung
- domänenspezifische Plattform → erweiterte Zielplattform
- DSL → Modellierungssprache im Problemraum
- Metamodell → Sprachkonstrukte einer DSL
- Domänen-Architektur

# Domänen-Architektur (domain architecture)

- beschreibt die Umgebung vom Modell zum Softwareprodukt
- beinhaltet
  - DSL/ Metamodell
  - Modelltransformation
  - Zielplattform

# Glossar

- Domäne → abgegrenzter Problemraum
- Modell → formalisiertes Wissen der Domäne
- Generator (M2T) → Generierung von Artefakten
- Transformator (M2M) → Modell – zu – Modell
- Generat → Ergebnis der Generierung
- Zielplattform → Laufzeitumgebung
- domänenspezifische Plattform → erweiterte Zielplattform
- DSL → Modellierungssprache im Problemraum
- Metamodell → Sprachkonstrukte einer DSL
- Domänen-Architektur → Umgebung Modell zu Softwareprodukt
- Software Systemfamilie

# Software Systemfamilie

- Programmgruppe mit gemeinsamen Eigenschaften
- hier: gemeinsame Domänen-Architektur

# Glossar

- **Domäne** → abgegrenzter Problemraum
- **Modell** → formalisiertes Wissen der Domäne
- **Generator (M2T)** → Generierung von Artefakten
- **Transformator (M2M)** → Modell – zu – Modell
- **Generat** → Ergebnis der Generierung
- **Zielplattform** → Laufzeitumgebung
- **domänenspezifische Plattform** → erweiterte Zielplattform
- **DSL** → Modellierungssprache im Problemraum
- **Metamodell** → Sprachkonstrukte einer DSL
- **Domänen-Architektur** → Umgebung Modell zu Softwareprodukt
- **Software Systemfamilie** → gemeinsame Domänen-Architektur

# Fragen ?