



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

AW1 Vortrag von Hauke Wittern

# Modellgetriebene Entwicklung von Pervasive Games

Entwicklung einer  
domänenspezifischen Sprache

# Agenda

---

- Motivation
- Pervasive Games
- Modellgetriebene Softwareentwicklung (MDSD)
- Herausforderungen bei der Entwicklung von Pervasive Games
- Potential von MDSD bei Pervasive Games
- Kategorisierung domänenspezifischer Sprachen
- Domänenspezifische Sprache für Pervasive Games
- Entwicklungsablauf
- Risiken

# Motivation

---

- Entwicklung von Pervasive Games ist komplex
- Spielfeld muss für jeden Spielort virtuell aufgebaut werden
- Komplexität muss reduziert werden
- Lösungsansatz:  
Einsatz modellgetriebener Softwareentwicklung  
und domänenspezifischer Sprachen

# Pervasive Games

- Spielerische Form der “mixed reality”
  - Virtuelle Welt
  - Reale Welt
- Basiert auf Pervasive Computing
- Häufig mit Mobile Computing Technologien



# Modellgetriebene Softwareentwicklung (MDSD)

---

- Modelle statt Code
- Generierung von Code aus den Modellen
- Domänenspezifische Sprachen (DSL)
  - Abstrakte Syntax: Metamodell
  - Konkrete Syntax

# Warum MDSD einsetzen?

---

- Hoher Abstraktionsgrad
- Einheitliche Architektur
- Interoperabilität
- Produktivitätssteigerung

# Herausforderungen bei der Entwicklung von Pervasive Games

---

- Positionsbestimmung der Spieler
- Kommunikation der Spieler untereinander und mit dem System
- Verwendung unterschiedlicher Geräte
- Entwicklung der Game Engine
- Modellierung des Spiels
- Orchestrierung und Überwachung eines konkreten Spiels

# Herausforderungen bei der Entwicklung von Pervasive Games

---

- Positionsbestimmung der Spieler → *Plattform*
- Kommunikation der Spieler untereinander und mit dem System → *Protokolle, Plattform*
- Verwendung unterschiedlicher Geräte → *Plattform, MDSD / Codegenerierung*
- Entwicklung der Game Engine → *Engine ist die Plattform*
- Modellierung des Spiels → *MDSD / DSL*
- Orchestrierung und Überwachung eines konkreten Spiels → *Tools der Plattform, (DSL?)*

# Potential von MDSD bei Pervasive Games

---

- Hoher Abstraktionsgrad erleichtert die Entwicklung
- Interoperabilität
  - Gerätespezifische Codegenerierung
  - Aber nur bis zu einem gewissen Grad
- Produktivitätssteigerung, da Änderungen zu erwarten sind
  
- Modelle als Diskussionsgrundlage

# Kategorisierung von DSLs

---

- Konkrete Syntax ist wichtig
  - Muss Domänenkonzepte klar und effizient ausdrücken
- Interne vs. externe DSL
- Grafische vs. textuelle DSL
- Wiederverwendung vs. Neuimplementierung

# DSL für Pervasive Games

---

- Grafische, externe DSL scheint angebracht zu sein (Diskussionsgrundlage!)
- UML mit Profilen?
  - ☺ Werkzeuge vorhanden
  - ☺ UML weit verbreitet
  - ☹ Unklare Semantik
- Benutzerdefinierte DSL?
  - ☺ Flexibel / keine Grenzen
  - ☺ Kann Konzepte optimal darstellen
  - ☹ Werkzeuge (Parser, Editoren) müssen selbst erstellt werden

# DSL für Pervasive Games

---

- Mehrere Abstraktionsebenen
  - Allgemeines Framework für Pervasive Games
  - Spezielle *Familien* von Spielen
- Verschiedene DSLs für die Abstraktionsebenen
  - Sprachen für Familien von Spielen und Framework
  - Modell-zu-Modell Transformation dazwischen
  - „End-Entwickler“ benutzt nur eine DSL!  
→ die Sprache der Produktlinie

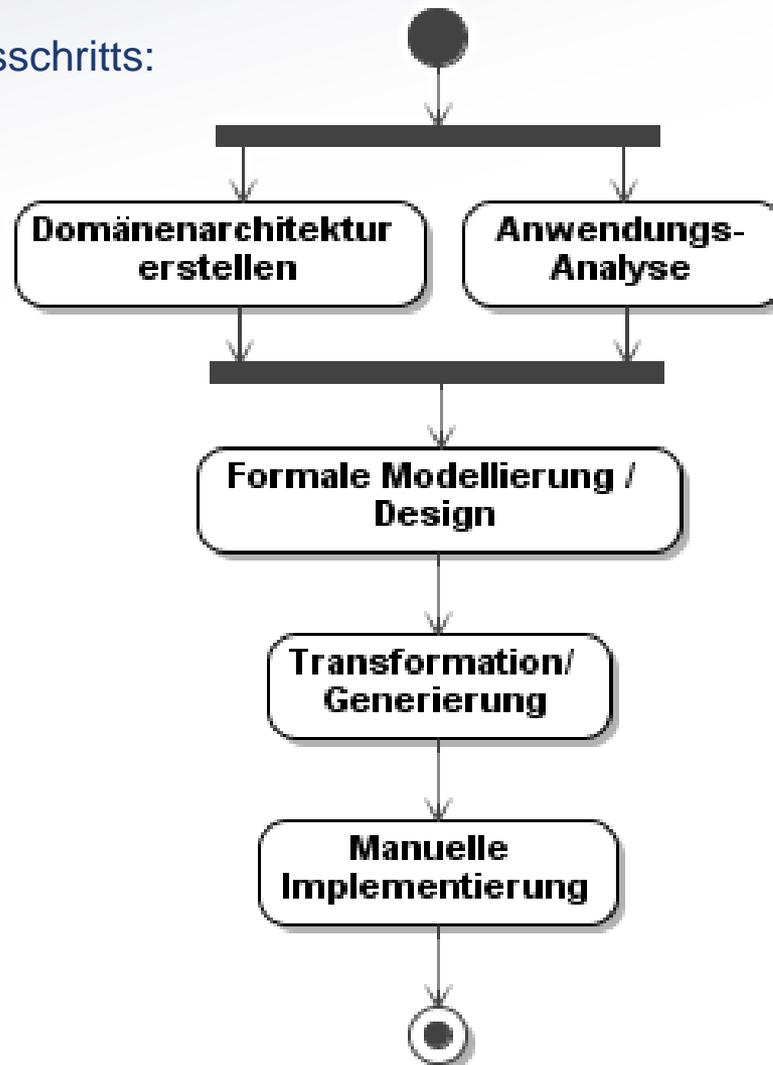
# DSL für Pervasive Games

---

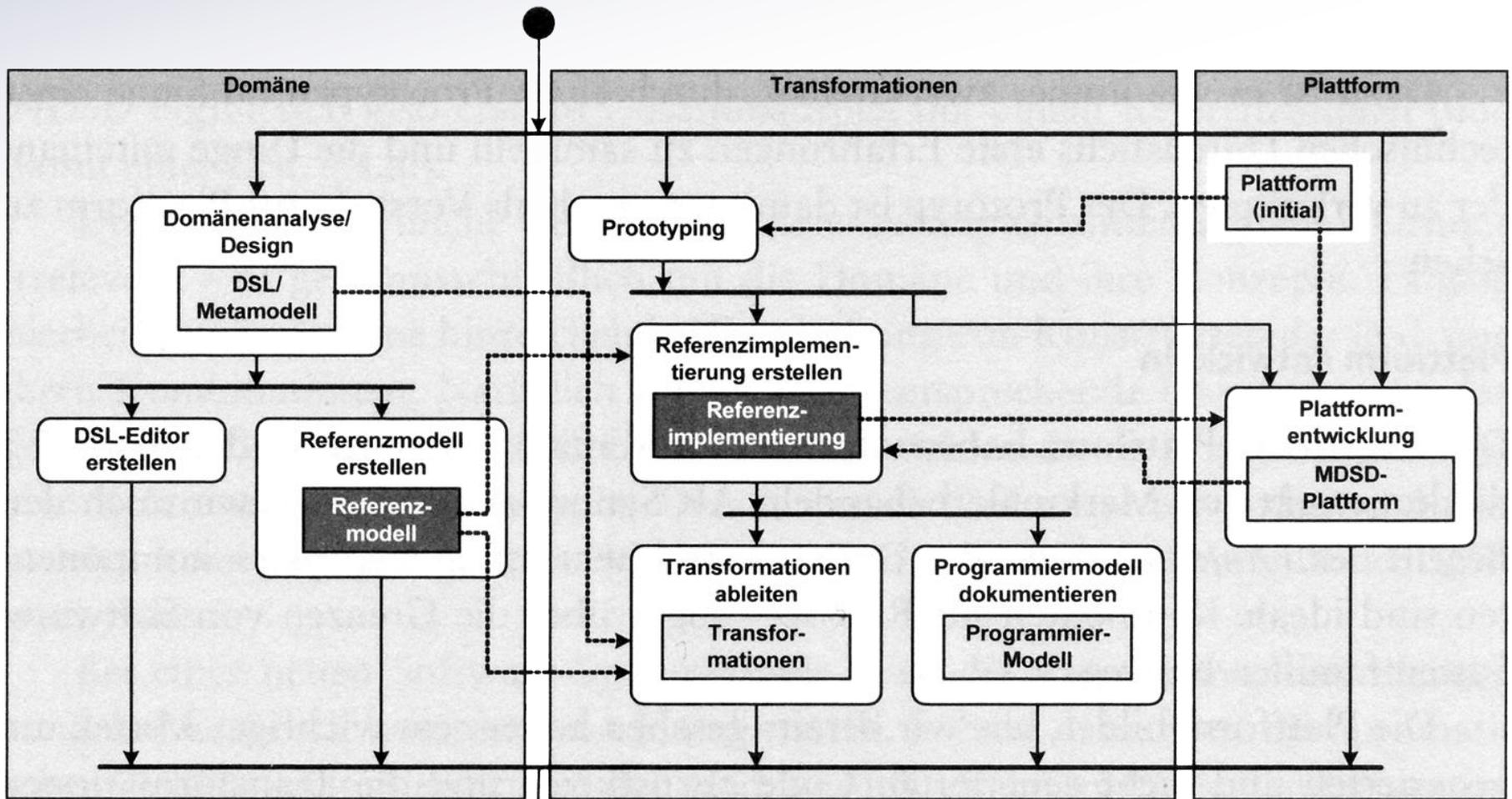
- DSL für Framework:
  - Grafisch und extern
  - UML Profile
- DSLs für Familien von Spielen:
  - Grafisch und extern
  - Benutzerdefinierte DSL wünschenswert
  - Aber zunächst UML Profile im Forschungsprojekt

# Entwicklungsablauf

Ablauf eines Iterationsschritts:



# Erstellung der Domänenarchitektur



# Risiken

---

- Syntax der DSL nicht angemessen
- Werkzeuge nicht vorhanden / nicht angemessen
- Größe der Iterationszyklen

# Zusammenfassung & Ausblick

---

- MDSD => Abstraktion!
- DSL mit Grafischer Syntax
- Zwei Abstraktionsebenen, Zwei DSLs

# Zusammenfassung & Ausblick

---

- Geplante Ergebnisse für Projekt, AW2
  - DSL für eine Familie von Pervasive Games, UML basiert
  - Referenzmodell
  - Referenzimplementierung
  - Modell-Transformationen
  - Framework / Plattform für Pervasive Games (Gemeinsames Projektergebnis)