



# Model Driven Software Development (MDSD)

---

## Model Driven Architecture (MDA<sup>®</sup>)

- Übersicht, Modell-Ebenen
- Beispiel-Notationen
- *Best Practices*
- Gutes Vorgehen
- Beispiel
  
- Literatur



MDS

- ▶ Übersicht
- ▶ Notation
- ▶ *Best Practices*
- ▶ Gutes Vorgehen
- ▶ Beispiel
- ▶ Literatur
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...

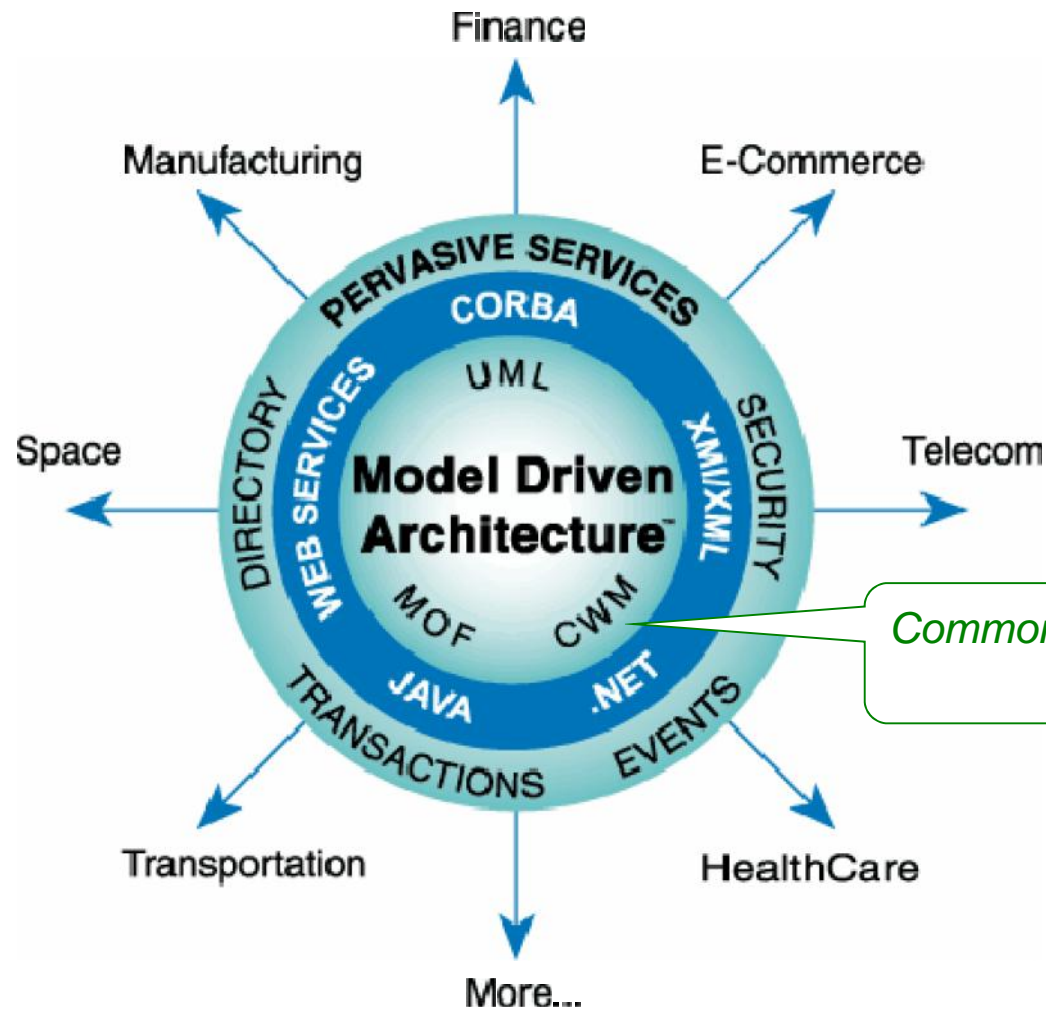
- MDD  
Model-Driven Development
- MDS  
Model-Driven Software Development
- MDA®  
Model-Driven Architecture,  
Warenzeichen der Object Management Group (OMG),  
[www.omg.org](http://www.omg.org)
- Gleiches Prinzip...



# Das Logo

MDS

- ▶ Übersicht
- ▶ Notation
- ▶ Best Practices
- ▶ Gutes Vorgehen
- ▶ Beispiel
- ▶ Literatur
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...



*Common Warehouse Meta Model*



# How Systems Will Be Built

## Das Ziel

MDS

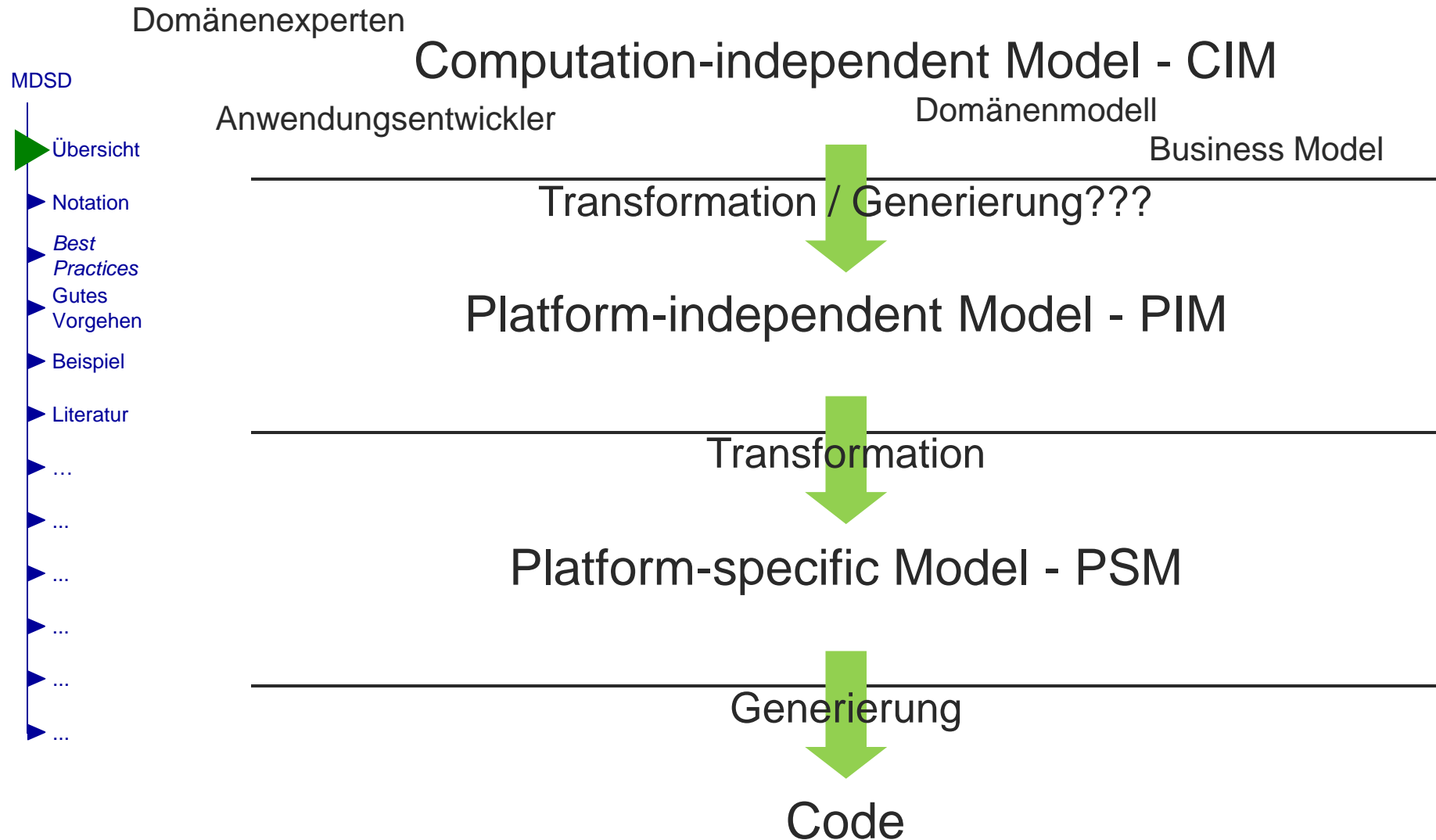
- ▶ Übersicht
- ▶ Notation
- ▶ Best Practices
- ▶ Gutes Vorgehen
- ▶ Beispiel
- ▶ Literatur
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...

OMG's Model Driven Architecture ® (MDA ®) provides an open, vendor-neutral approach to the challenge of business and technology change. Based on OMG's established standards, the MDA separates business and application logic from underlying platform technology. Platform-independent models of an application or integrated system's business functionality and behavior, built using UML and the other associated OMG modeling standards, can be realized through the MDA on virtually any platform, open or proprietary, including Web Services, .NET, CORBA ®, J2EE, and others. These platform-independent models document the business functionality and behavior of an application separate from the technology-specific code that implements it, insulating the core of the application from technology and its relentless churn cycle while enabling interoperability both within and across platform boundaries.

No longer tied to each other, the business and technical aspects of an application or integrated system can each evolve at its own pace - business logic responding to business need, and technology taking advantage of new developments - as the business requires.



# Modell-Ebenen

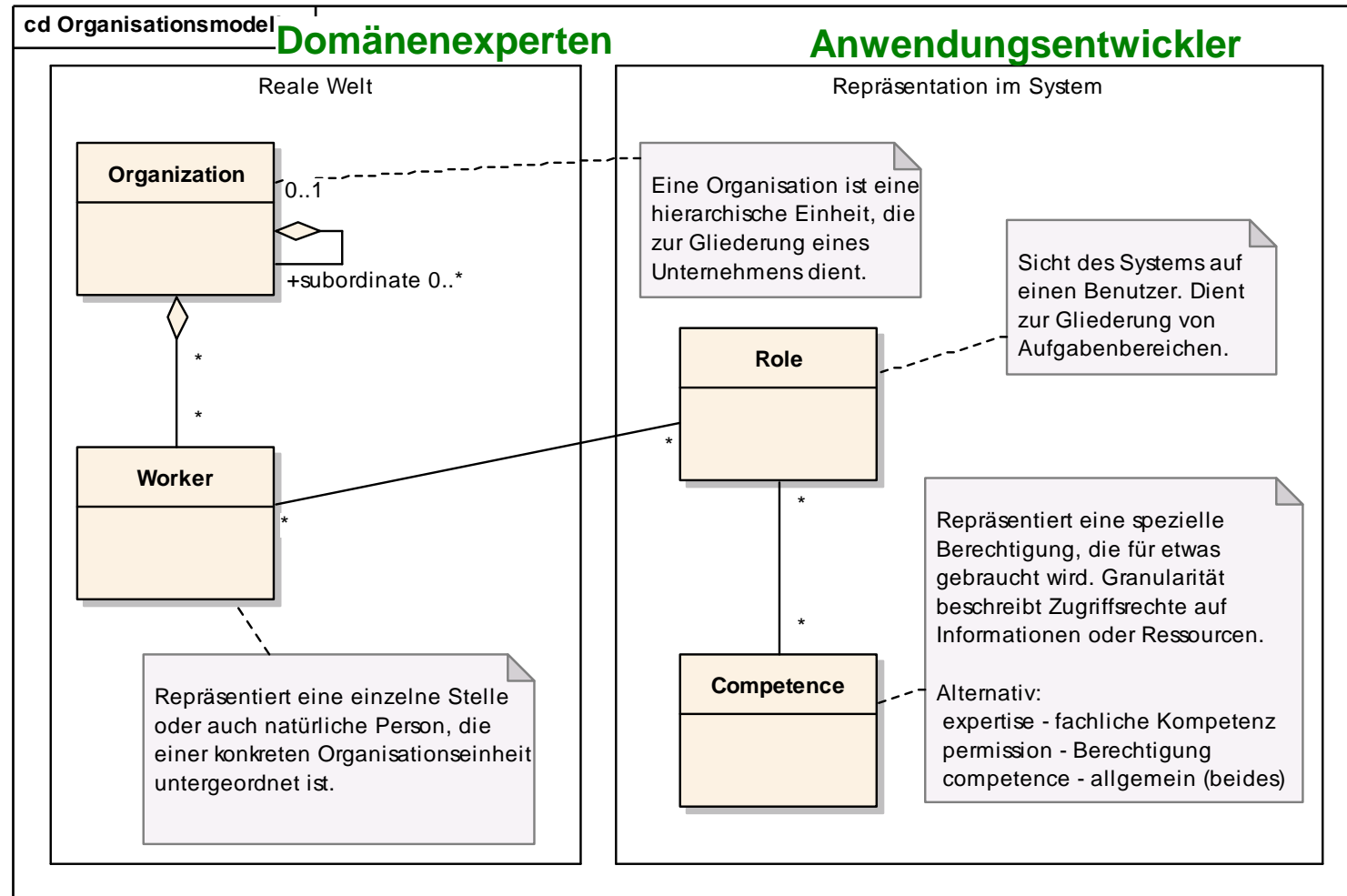




# CIM – Unterschiedliche Sichten: Domänenexperten <-> Anwendungsentwickler

MDSD

- ▶ Übersicht
- ▶ Notation
- ▶ Best Practices
- ▶ Gutes Vorgehen
- ▶ Beispiel
- ▶ Literatur
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...

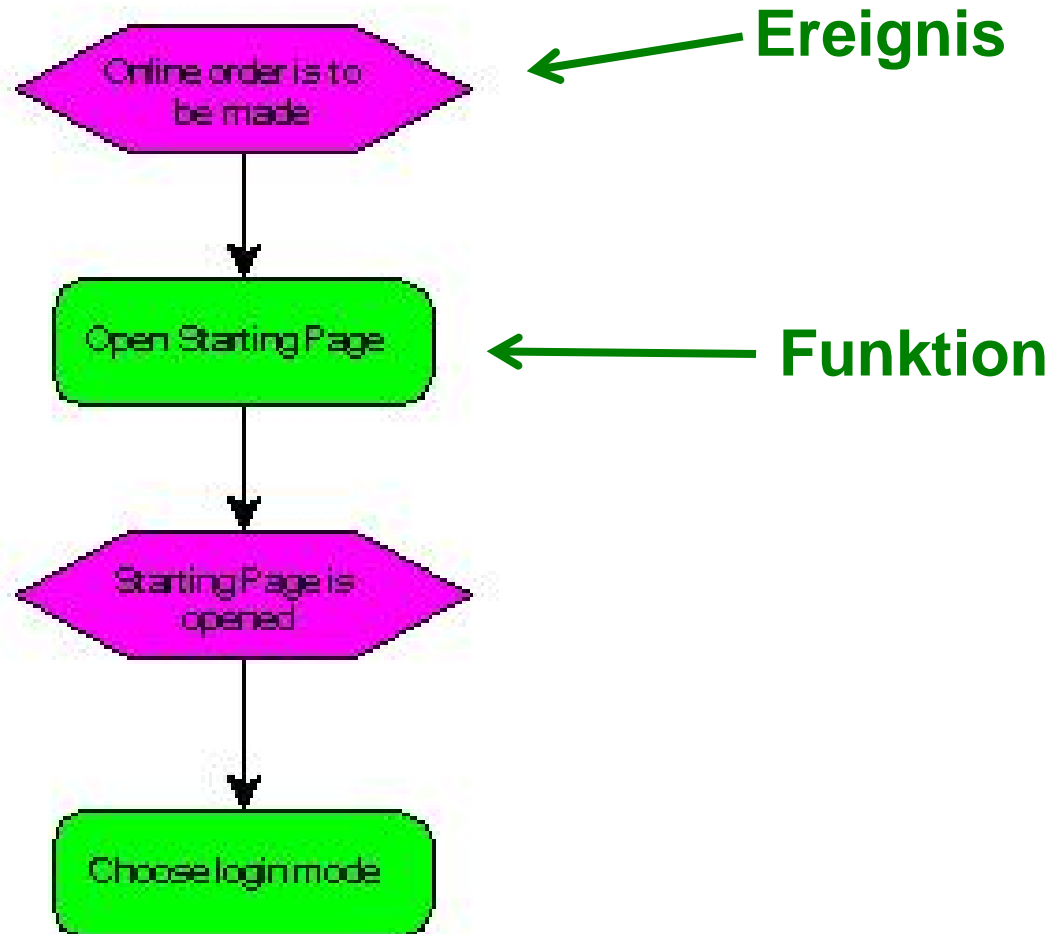




# Mögliche CIM-Notation: Ereignisgesteuerte Prozessketten (EPK, ARIS)

MDS

- Übersicht
- Notation**
- Best Practices
- Gutes Vorgehen
- Beispiel
- Literatur
- ...
- ...
- ...
- ...
- ...
- ...

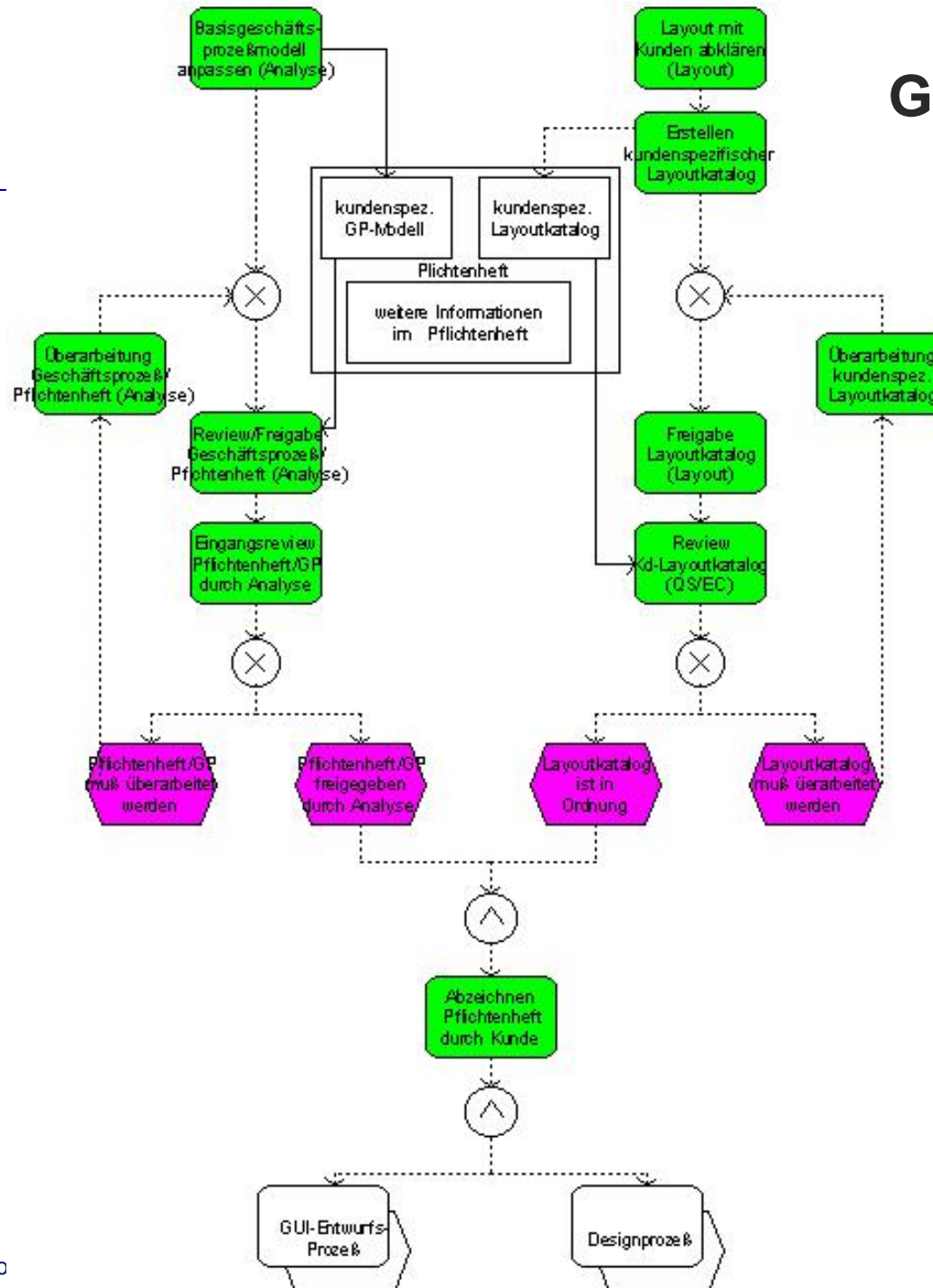




# Geschäftsprozesse (ARIS)

MDSD

- Übersicht
- Notation
- Best Practices
- Gutes Vorgehen
- Beispiel
- Literatur
- ...
- ...
- ...
- ...
- ...
- ...
- ...





# PIM



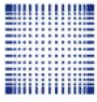
Anwendungsentwickler

MDS

- ▶ Übersicht
- ▶ Notation
- ▶ *Best Practices*
- ▶ Gutes Vorgehen
- ▶ Beispiel
- ▶ Literatur
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...

- Umsetzung des CIM Problem → Lösung
  - Vollständige Beschreibung der Funktionalität (→ Datenstrukturen etc.)
  - Keine Plattform-Spezifika
- ≈ A-Architektur (Anwendungsarchitektur)

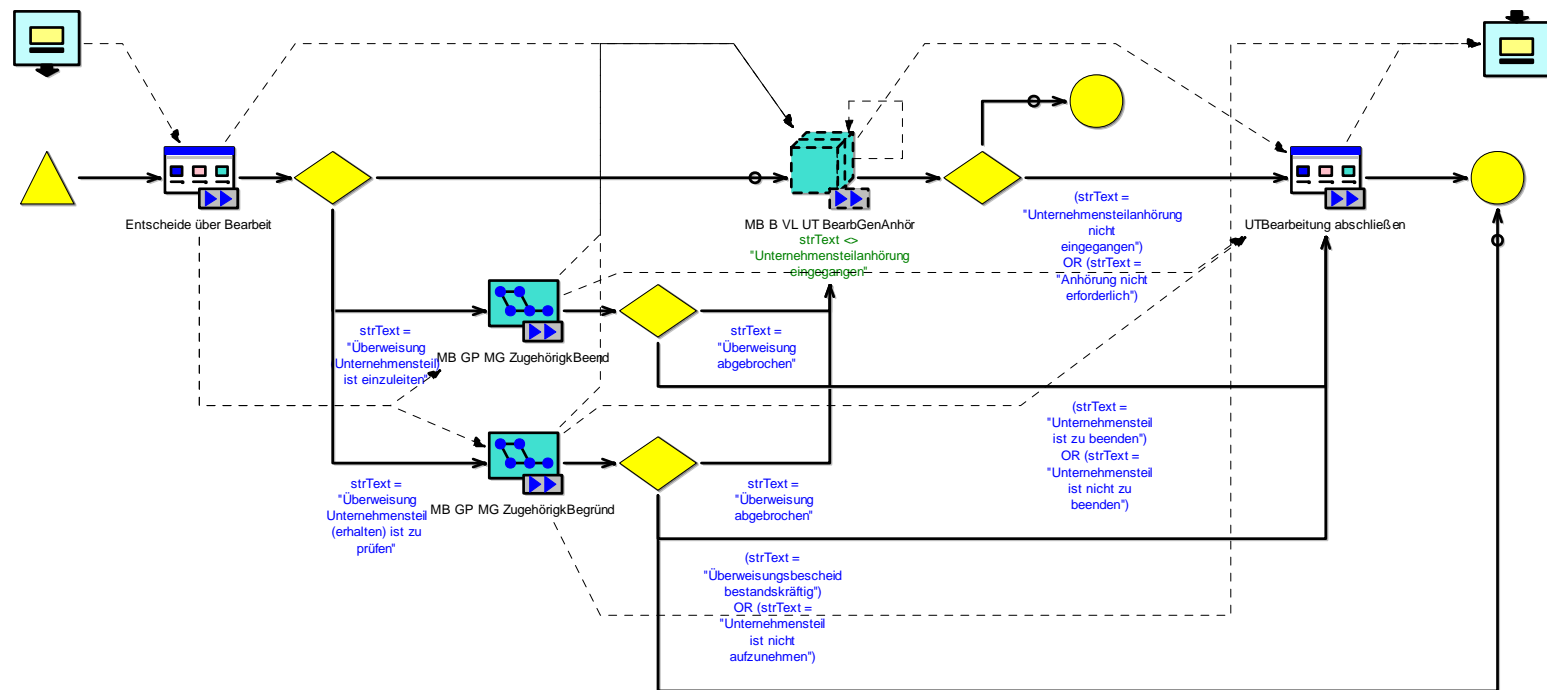
[Siedersleben]



# Mögliche PIM-Notation: Workflow (ADONIS)

MDS

- ▶ Übersicht
- ▶ Notation
- ▶ Best Practices
- ▶ Gutes Vorgehen
- ▶ Beispiel
- ▶ Literatur
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...

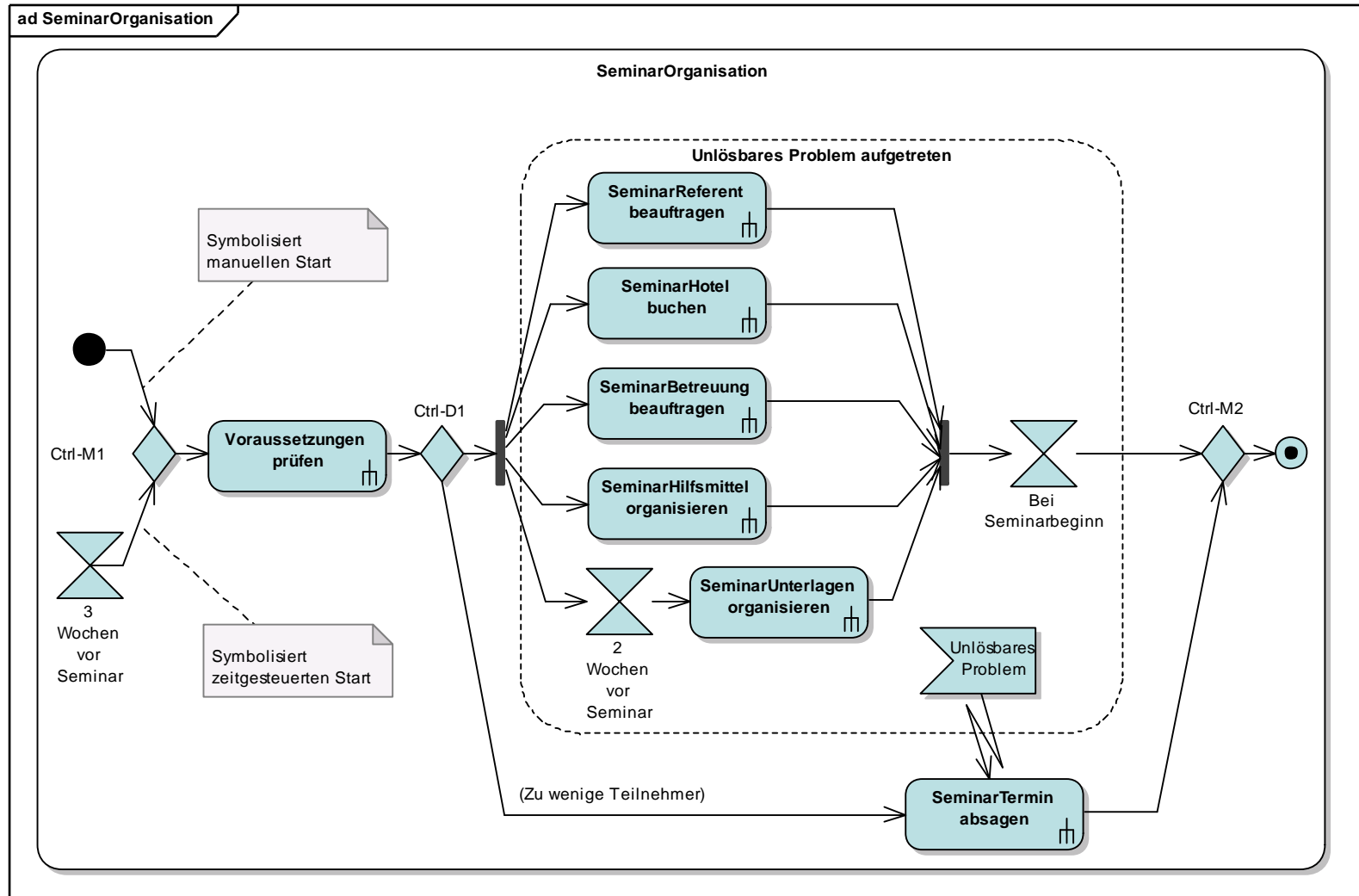




# Mögliche PIM-Notation: Aktivitätsdiagramm

MDSD

- ▶ Übersicht
- ▶ Notation
- ▶ Best Practices
- ▶ Gutes Vorgehen
- ▶ Beispiel
- ▶ Literatur
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...





MDS

- ▶ Übersicht
- ▶ Notation
- ▶ Best Practices
- ▶ Gutes Vorgehen
- ▶ Beispiel
- ▶ Literatur
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...

- Definierte Transformationsregeln:  
automatische *Transformation* PIM → PSM
  - Basis für die spätere Quellcode-  
*Generierung*  
→ plattform-spezifisch
- ≈ Hinzufügen der T-Architektur zur  
A-Architektur [Siedersleben]



# Abkürzungen und ihre Verwendung

MDSD

- ▶ Übersicht
- ▶ Notation
- ▶ *Best Practices*
- ▶ Gutes Vorgehen
- ▶ Beispiel
- ▶ Literatur
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...

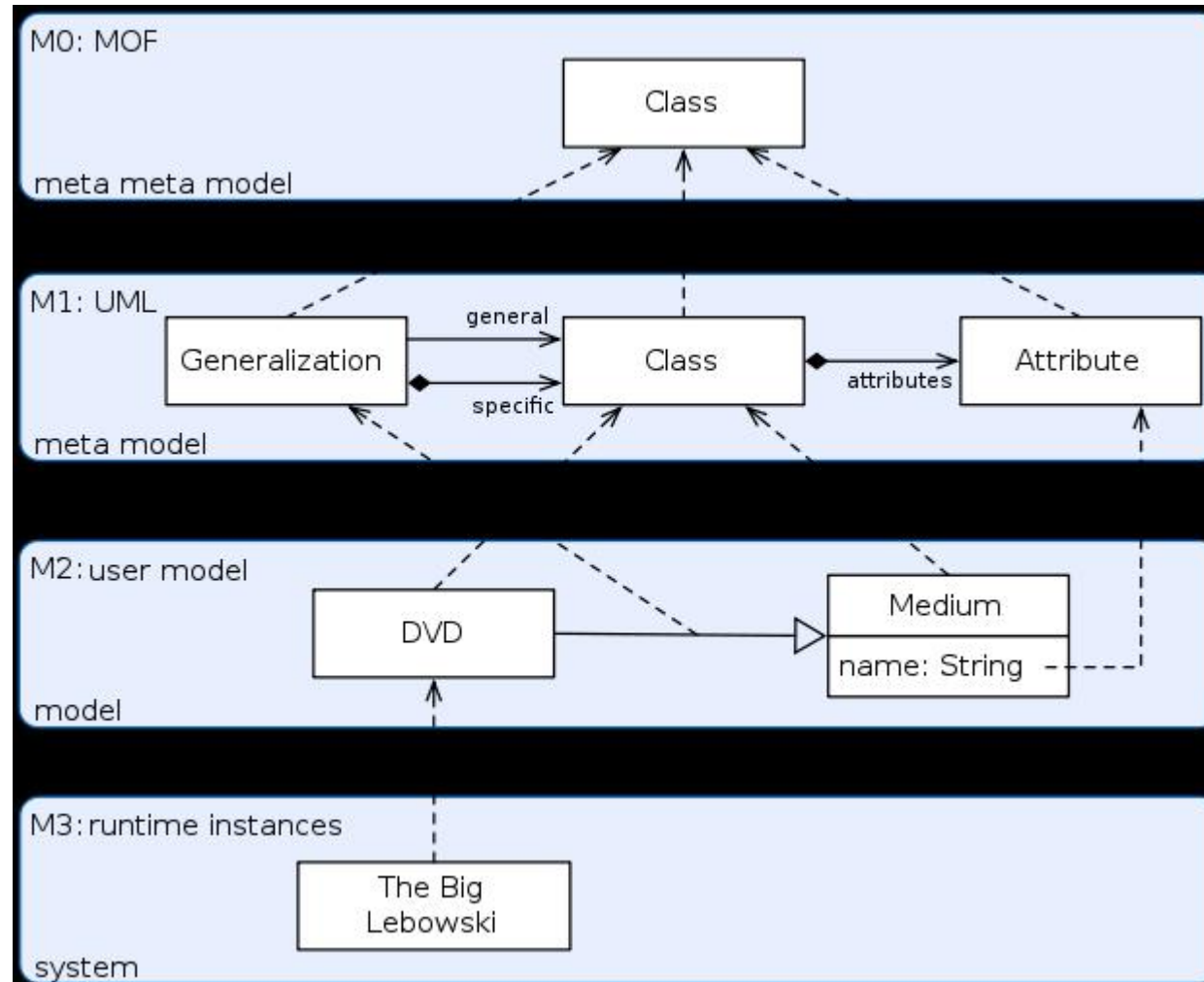
- UML 2 – Unified Modeling Language
  - Modellierung, Spezifikation
  - (Visualisierung, Dokumentation)
- MOF – Meta Object Facility
  - Meta-Sprache:  
Vokabular zur Definition von Modellierungssprachen (z.B., aber nicht nur UML)
  - Essential MOF, Complete MOF



# Veranschaulichung aus Wikipedia (leider anders herum nummeriert!)

MDSD

- ▶ Übersicht
- ▶ Notation
- ▶ Best Practices
- ▶ Gutes Vorgehen
- ▶ Beispiel
- ▶ Literatur
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...





# Abkürzungen und ihre Verwendung

MDSD

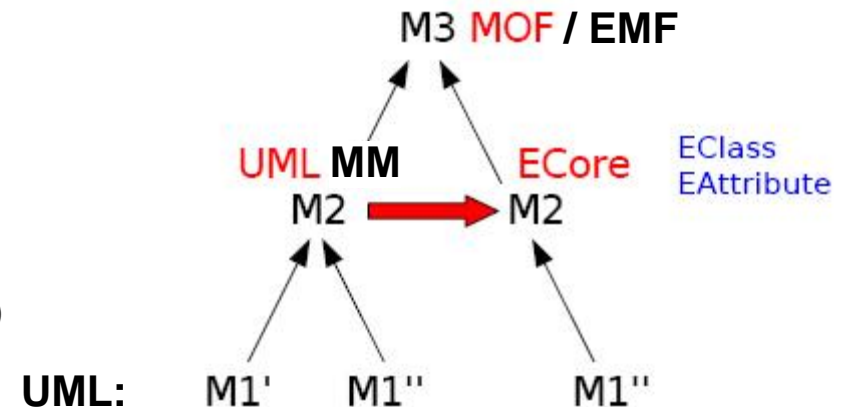
- ▶ Übersicht
- ▶ Notation
- ▶ Best Practices Gutes Vorgehen
- ▶ Beispiel
- ▶ Literatur
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...

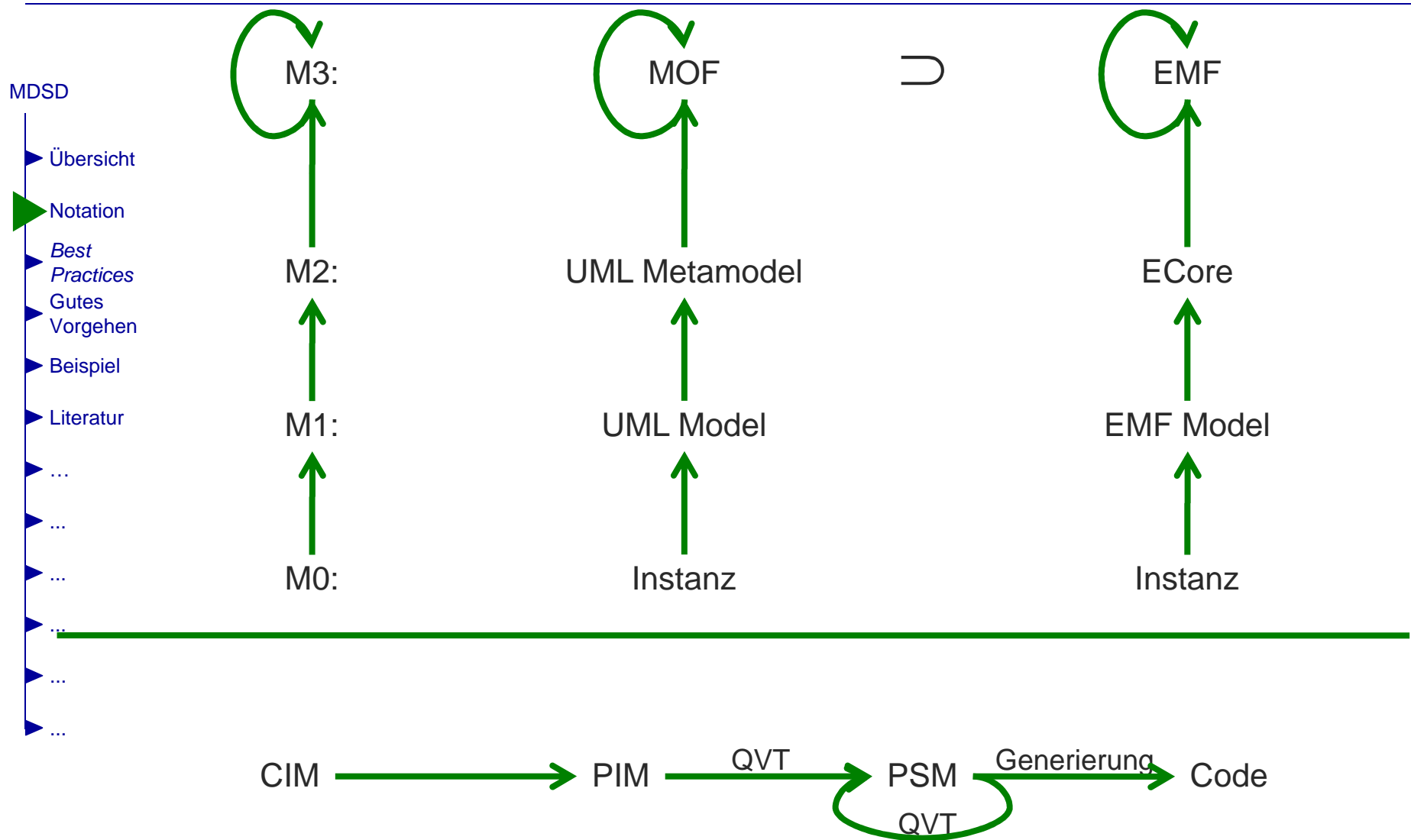
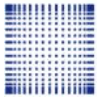
- UML 2 – Unified Modeling Language
  - Modellierung, Spezifikation
  - (Visualisierung, Dokumentation)
- MOF – Meta Object Facility
  - Meta-Sprache:  
Vokabular zur Definition von Modellierungssprachen (z.B., aber nicht nur UML)
  - Essential MOF, Complete MOF
- XMI – XML Metadata Interchange
  - Standardisiertes Austauschformat für alle MOF-basierten Sprachen

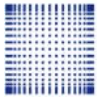
→ Kombination:

- "Einfache" Definition domänenspezifischer (Modell-) Sprachen
- Benutzung existierender (UML-) Modellierungstools

- Alternative (Eclipse-basiert):
  - EMF (Eclipse Modeling Framework; "relevante" Teilmenge, Alternative zu MOF)
  - ECore (Alternative zu UML2 Metamodel)



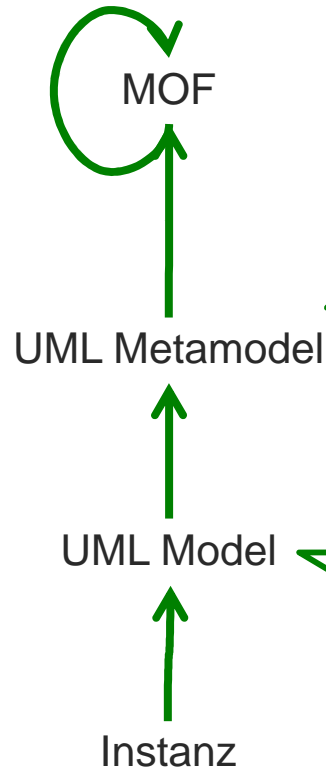




# Schwergewichtige und leichtgewichtige Ansätze

MDS

- Übersicht
- Notation
- Best Practices
- Gutes Vorgehen
- Beispiel
- Literatur
- ...
- ...
- ...
- ...
- ...
- ...
- ...



Mit Elementen der MOF neue Elemente vergleichbar zu "Klassen", "Attributen", "Vererbung" etc. definieren

Tools: basierend auf ECore etc.

Mit Stereotypen für existierende Metamodell-Elemente

Tools: Standard-UML



# Umsetzung verschiedener Best Practices

MDS

- ▶ Übersicht
- ▶ Notation
- ▶ **Best Practices**
- ▶ Gutes Vorgehen
- ▶ Beispiel
- ▶ Literatur
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...

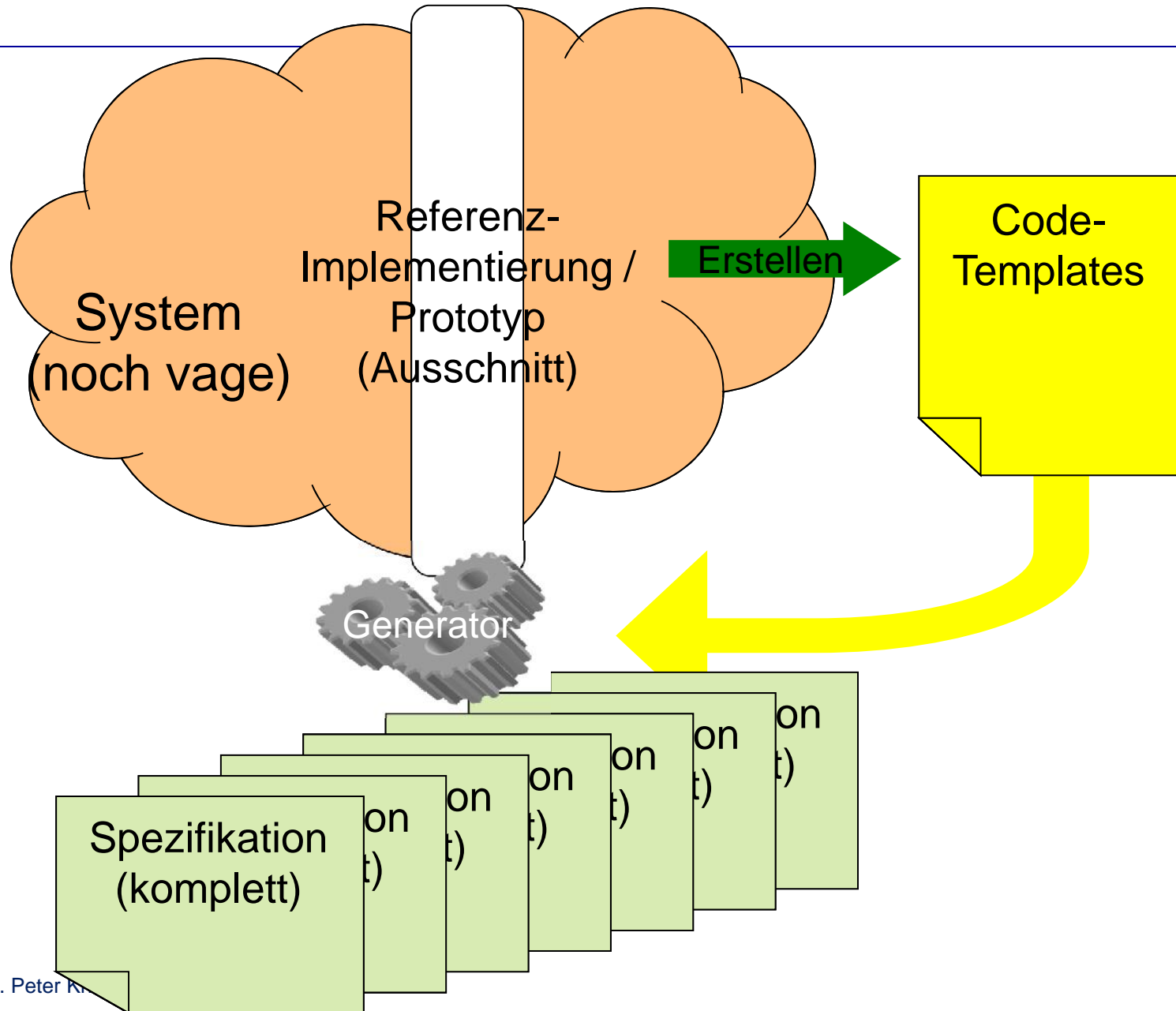
- Single Source of Information:  
die Modelle  
(aber auch Single Point of Failure: die Generatoren)
  - Konsistenz bei der Umsetzung gleicher/ähnlicher Konzepte  
(*architectural texture*):  
Generatoren
  - Trennen von A-Architektur und T-Architektur
- "Alles, was ich mehr als drei Mal gemacht habe (*machen werde*) sollte ein Generator übernehmen"



# Gutes Vorgehen

MDS

- ▶ Übersicht
- ▶ Notation
- ▶ Best Practices
- ▶ Gutes Vorgehen
- ▶ Beispiel
- ▶ Literatur
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...

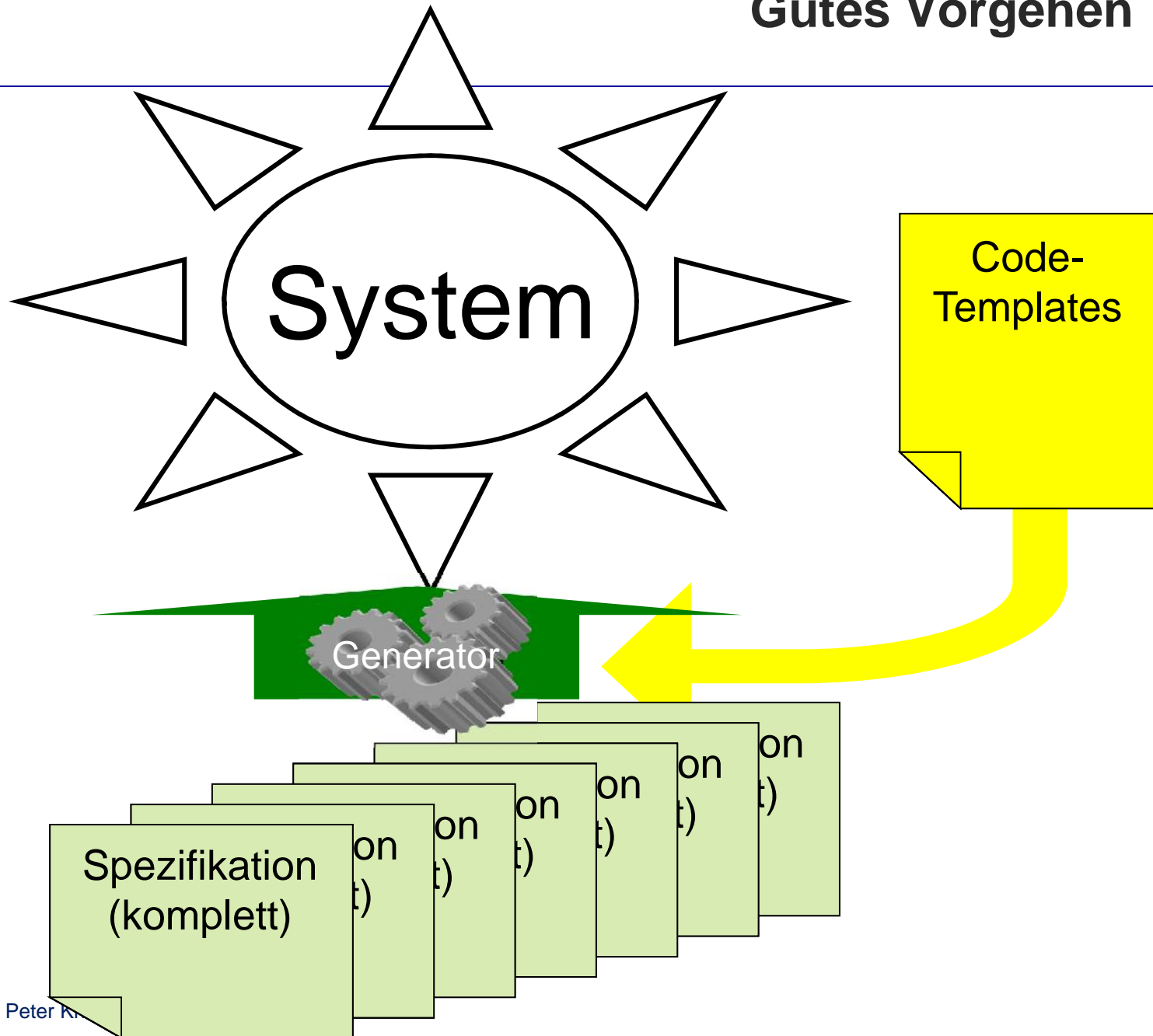




# Gutes Vorgehen

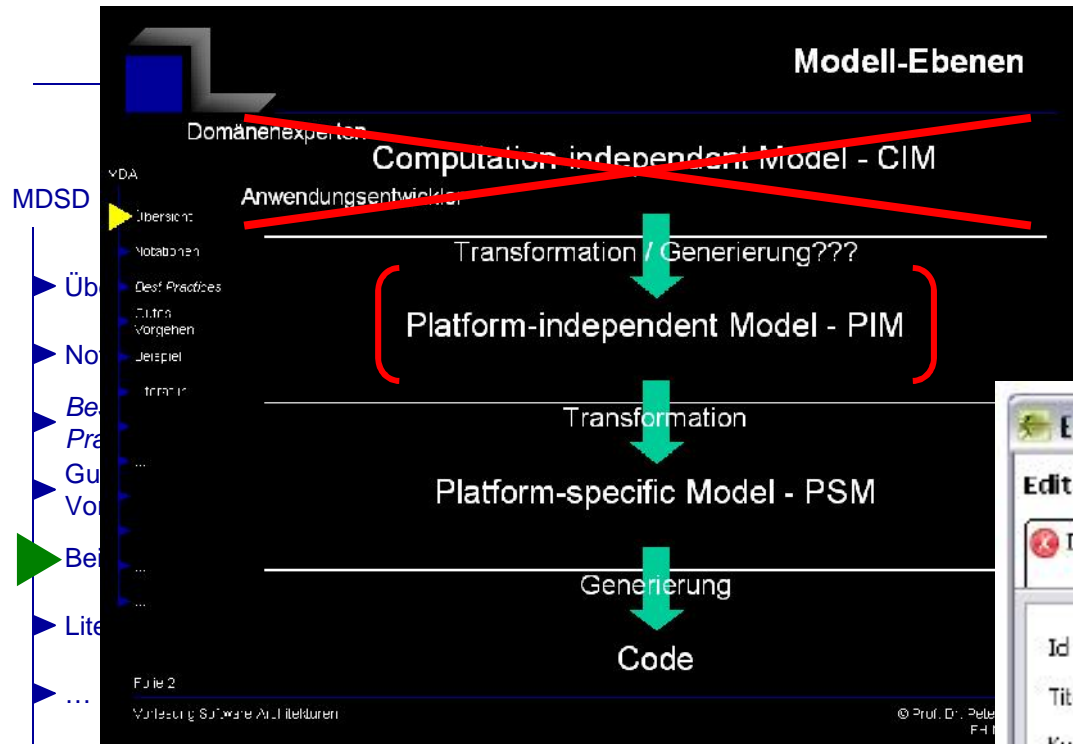
MDS

- ▶ Übersicht
- ▶ Notation
- ▶ Best Practices
- ▶ Gutes Vorgehen
- ▶ Beispiel
- ▶ Literatur
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...





# Beispiel



**Edit TitelVO**

Edit a TitelVO.

Invalid Titel!

Id	2
Titel	
Kurzbeschreibung	Principles, Patterns and Plug-Ins
ISBN	0-321-20575-8
CopyrightJahr	2004
Autor1	Erich Gamma
Autor2	Kent Beck
Verlag	Addison Wesley
Thema	Client-Entwicklung

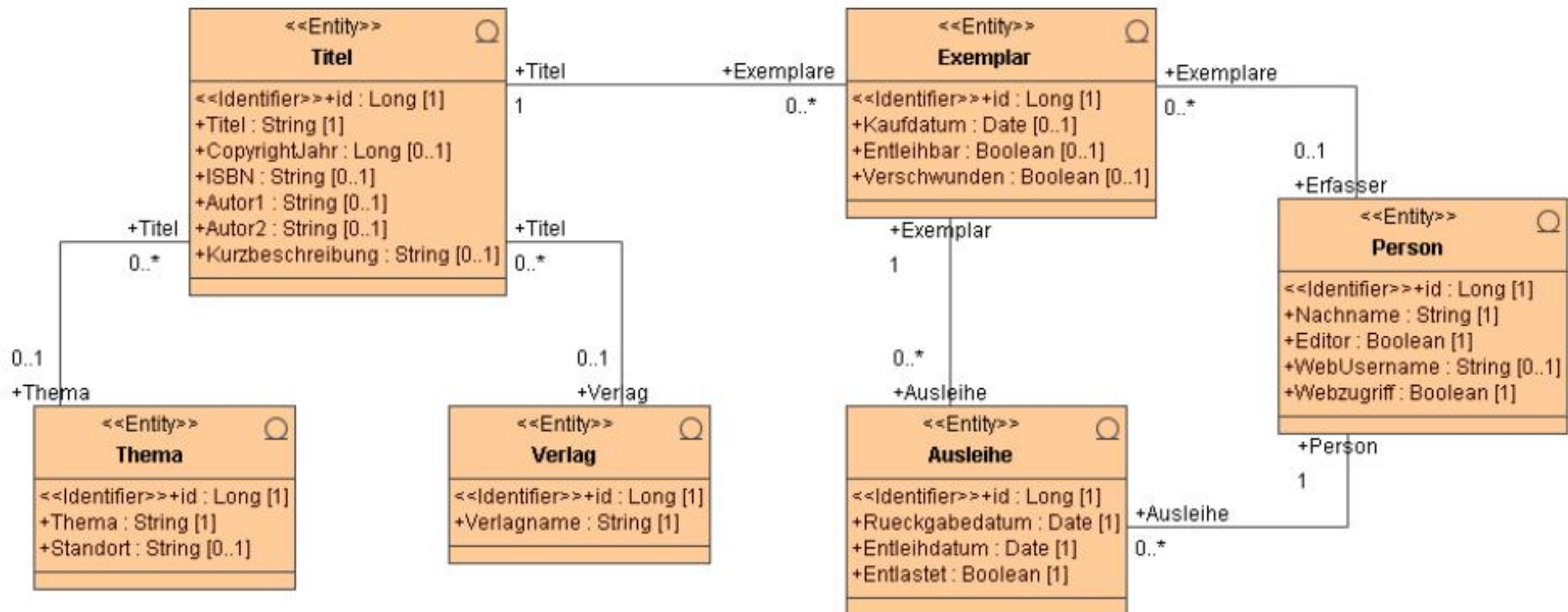
Finish Cancel

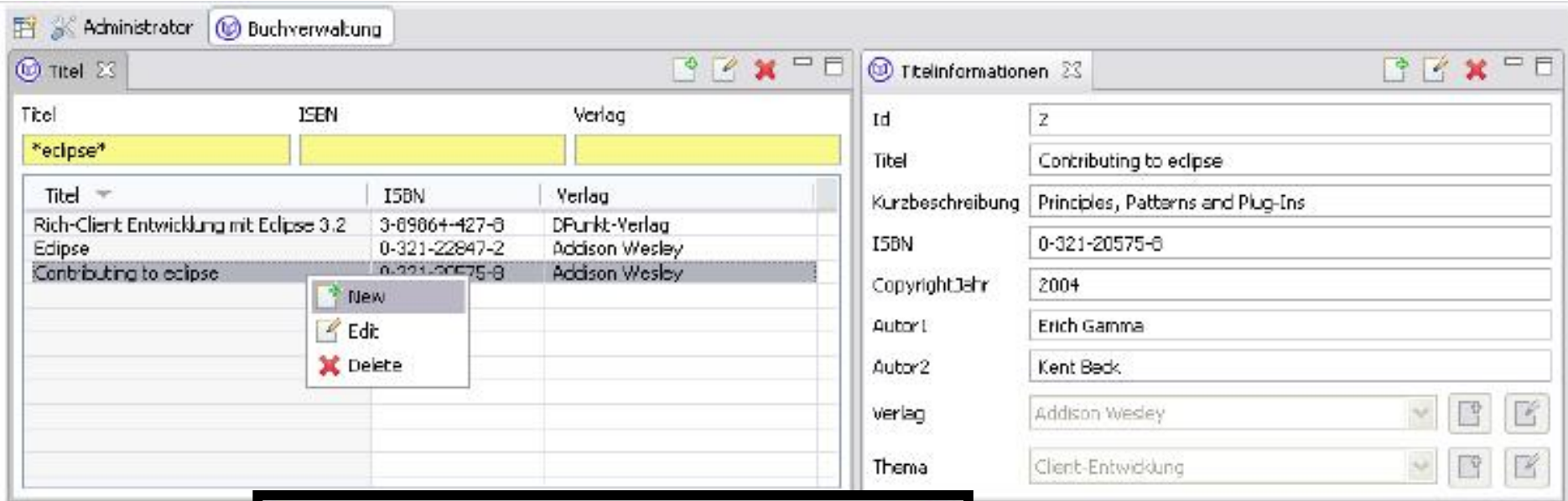


# Objekte des Geschäftsmodells

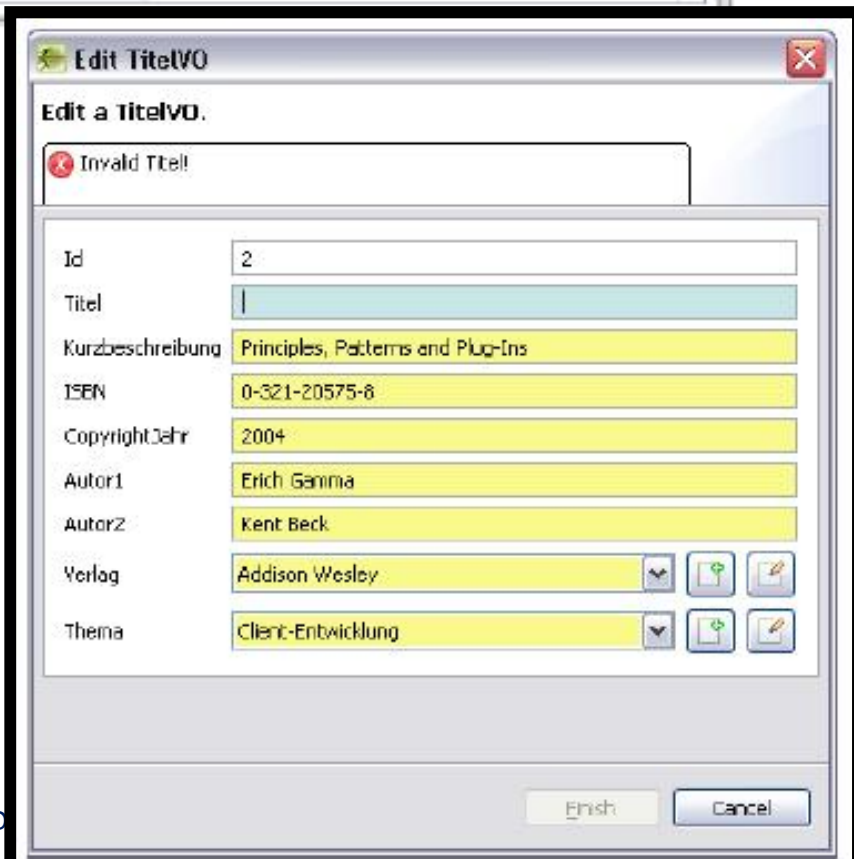
MDS

Übersicht





- Vorgehen
- ▶ Beispiel
- ▶ Literatur
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...



**Das Ergebnis**



## openArchitectureware has moved to the Eclipse Modeling Project

If you are looking for oAW, the MDA/MDD generator framework implemented in Java(TM) please note that its components (Workflow Engine, Xpand, Xtend, Check and Xtext) are no longer under active development at this site but have moved to Eclipse.org. As our [letter of intent](#) describes, these components are now available as part of the [Eclipse Modeling Project](#). Since moving to Eclipse, we have implemented a number of exciting new features and - with the help of our active community - fixed a number of bugs.

At Eclipse, we established a [working group](#) that guides you through the migration of existing projects. The working group provides Eclipse distributions that contain all components of oAW. The [newsgroup](#) supports you with any issue and helps you to find the proper place at Eclipse to ask your questions.

### Still looking for oAW 4.3.1 ?

We understand that the migration of large projects can be a quite an effort. Therefore, we will still provide the existing downloads and documentation for oAW 4.3.1. Please note that the [old forums](#) and other resources will become outdated over time.

[Former Version \(4.3.1\)](#)

### Need help migrating?

Beside the existing migration documentation [itemis offers professional support](#) and will gladly help you with the migration.

Copyright © 2009 openArchitectureWare.org

The components you are looking for are now part of the [Eclipse Modeling Project](#). You will find comprehensive documentation and support at the appropriate Eclipse websites:

- [Xtext Website \(Forum\)](#)
- [Xpand/Xtend/Check Website \(Forum\)](#)
- [mvn website \(Forum\)](#)

Also, for newcomers of these components at Eclipse, please visit the

[Working Group at Eclipse](#)

or read the [letter of intent](#).


MD

Eclipse Modeling Project - Mozilla Firefox  
http://www.eclipse.org/modeling/ umlc meta model umle

Home Downloads Users Members Committers Resources Projects About Us

## Eclipse Modeling Project

The Eclipse Modeling Project focuses on the evolution and promotion of model-based development technologies within the Eclipse community by providing a unified set of modeling frameworks, tooling, and standards implementations.



The Modeling Project charter is posted [here](#) and inherits from the [Eclipse Standard Top-Level Charter v1.0](#).

### Abstract Syntax Development

- » [Eclipse Modeling Framework \(EMF\)](#) : a modeling framework and code generation facility for building tools and other applications based on a structured data model.
- » [CDO](#) : a technology for distributed shared EMF models and a fast server-based O/R mapping solution. With CDO you can easily enhance your existing models in such a way that saving a resource transparently commits the applied changes to a relational database.
- » [Net4j](#) : an extensible client-server system based on the Eclipse Runtime and the Spring Framework. You can easily extend the protocol stack with Eclipse plugins that provide new transport or application protocols.
- » [Teneo](#) : a database persistency solution for EMF using Hibernate or EclipseLink. It supports automatic creation of EMF to Relational Mappings and the related database schemas.
- » [Model Query \(MQ\)](#) : facilitates the process of search and retrieval of model elements of interest in a flexible yet controlled and structured manner.
- » [Model Transaction \(MT\)](#) : provides a model management layer built on top of EMF for managing EMF resources.
- » [Validation Framework \(VF\)](#) : provides model constraint definition, traversal, and evaluation for EMF model validation.

### Related links

- [Modeling Project Wiki](#)
- [Modeling Corner](#)
- [Helios, Galileo, Ganymede, Europa, Callisto](#) - see 'Models and Model Development Tools' category.
- [Who's Who in Modeling?](#)

### News

- [All-REQS All-In-One Galileo SR1 Modeling Tools Package](#) is available.
- [Domain Specific Modeling demo](#) viewlets from [Eclipse Summit Europe 2007](#) are posted.

MD

Find:  Next Previous Highlight all Match case

Done

AndroMDA.org - Home - Mozilla

File Edit View Go Bookmarks Tools Window Help

Back Forward Reload Stop <http://www.andromda.org/> Search Print

Home Bookmarks mozilla.org mozillaZine mozdev.org

# AndroMDA

Home News Downloads Documentation Support Search Developers **AndroMDA.com**

## Cutting Edge MDSD/MDA Toolkit

- Flexible
- Simple
- Ready-to-use
- Customizable
- Powerful

AndroMDA is one of the most powerful Open Source MDA Generators on the planet. It is used all over the world for everything from simple CRUD applications to complex enterprise applications. AndroMDA comes with an array of ready-made cartridges for common architectures like **Spring**, **EJB**, **.NET**, **Hibernate**, **Struts** and even more. The **project wizard** gets you started in no time. An **Eclipse integration** is in the works and will be finished soon. If you have a custom architecture, AndroMDA can also help you - by writing a so-called **cartridge**, you can customize AndroMDA to fit your needs.

[Demos](#) [Download](#) [Details](#)

### About AndroMDA

- [What is AndroMDA](#)
- [About Us](#)
- [Demos](#)
- [FAQs](#)
- [In the Press](#)
- [Events](#)
- [UML Tools](#)
- [Powered by AndroMDA](#)
- [Companies using AndroMDA](#)
- [What the users say](#)
- [Polls](#)

### Welcome to AndroMDA!

## News: New AndroMDA trainings starting in August!

Events

Monday, 28 May 2007

A new series of AndroMDA training courses is available, for both AndroMDA 3.x and AndroMDA 4.x! You can find the course descriptions and the list of dates where the courses will be given, both at <http://www.andromda.com/>

Please register online, there, too.

## AndroMDA 4.0-M1-SNAPSHOT for preview



## MDSD

- ▶ Übersicht
- ▶ Notation
- ▶ *Best Practices*
- ▶ Gutes Vorgehen
- ▶ Beispiel
- ▶ **Literatur**
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...

- OMG: [www.omg.org/mda](http://www.omg.org/mda)
- openArchitectureWare: [www.eclipse.org/modeling](http://www.eclipse.org/modeling)
- androMDA: [www.andromda.org](http://www.andromda.org)
- Acceleo: [www.eclipse.org/acceleo](http://www.eclipse.org/acceleo)
- A. Gordt: Iterative Einführung modellgetriebener Softwareentwicklungsprozesse in geschäftskritischen IT-Projekten, FH Mannheim, Diplomarbeit, 2005
- C. Löwer: Generierung von Eclipse RCP-Clients mittels AndroMDA, HS Mannheim, Diplomarbeit, 2006
- A. Pilch: Evaluation modellgetriebener Softwareentwicklungsmethoden im Kontext des J2EE-basierten "Enterprise Java Programming Model" (EJPM) Frameworks von John Deere
- C. Rensmann: Konzept zur Erstellung standardisierter Software mittels modellgetriebener Architektur am Beispiel des "Enterprise Java Programming Model 2.0" von John Deere