Fakultät Informatik, Institut für Software- und Multimediatechnik, Lehrstuhl für Softwaretechnologie
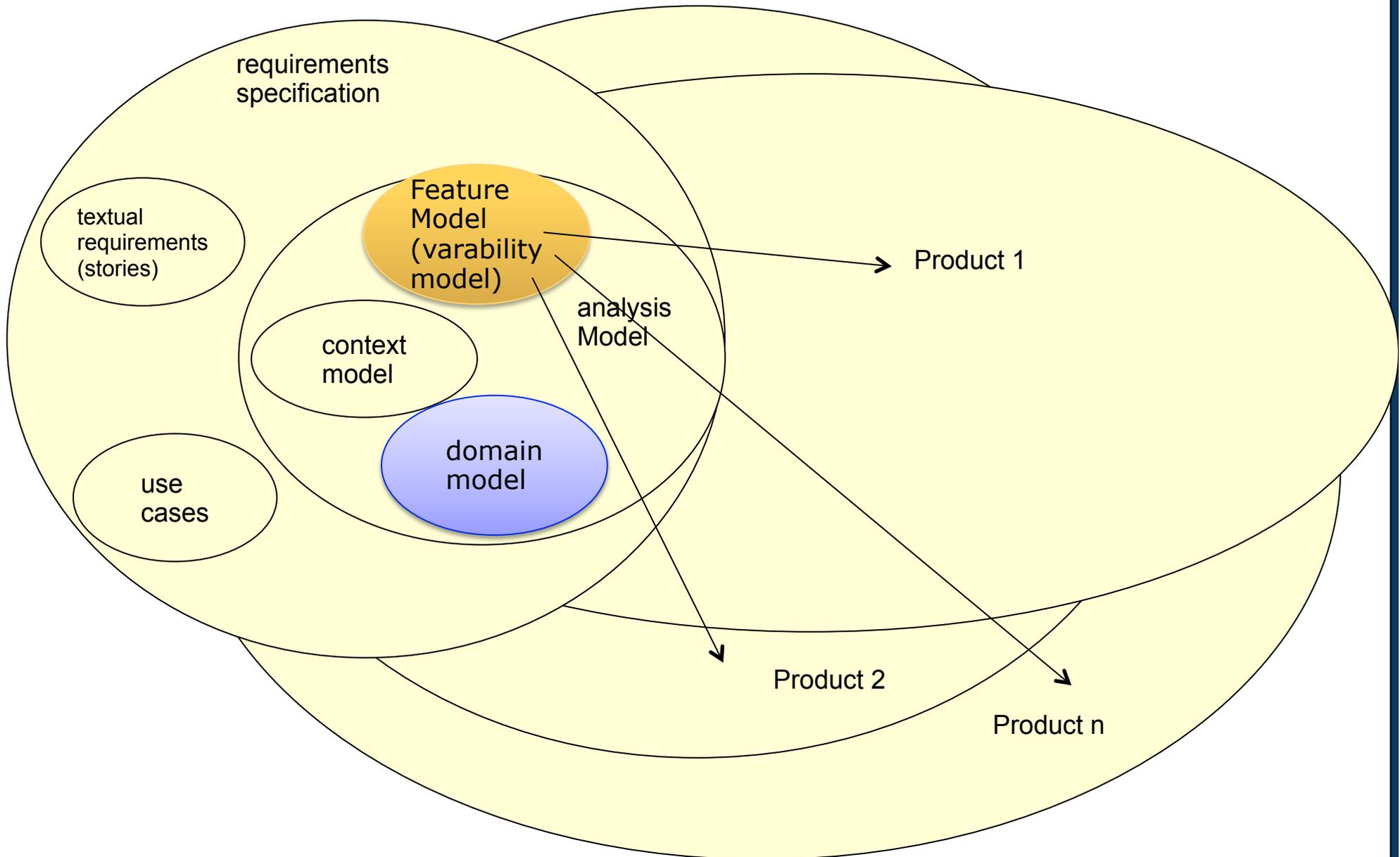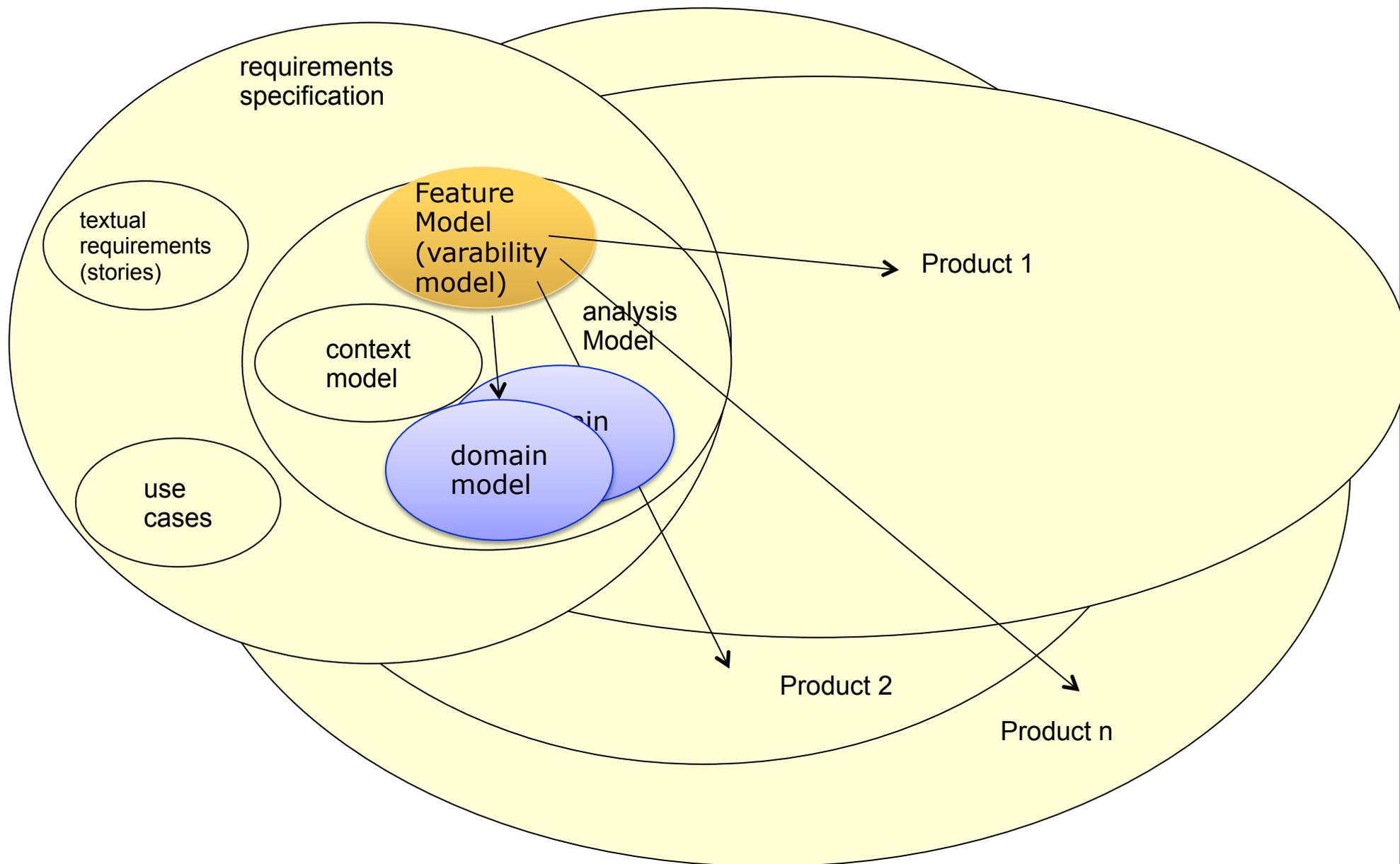
# 32) Domain Models and Software Product Line Engineering (SPLC)

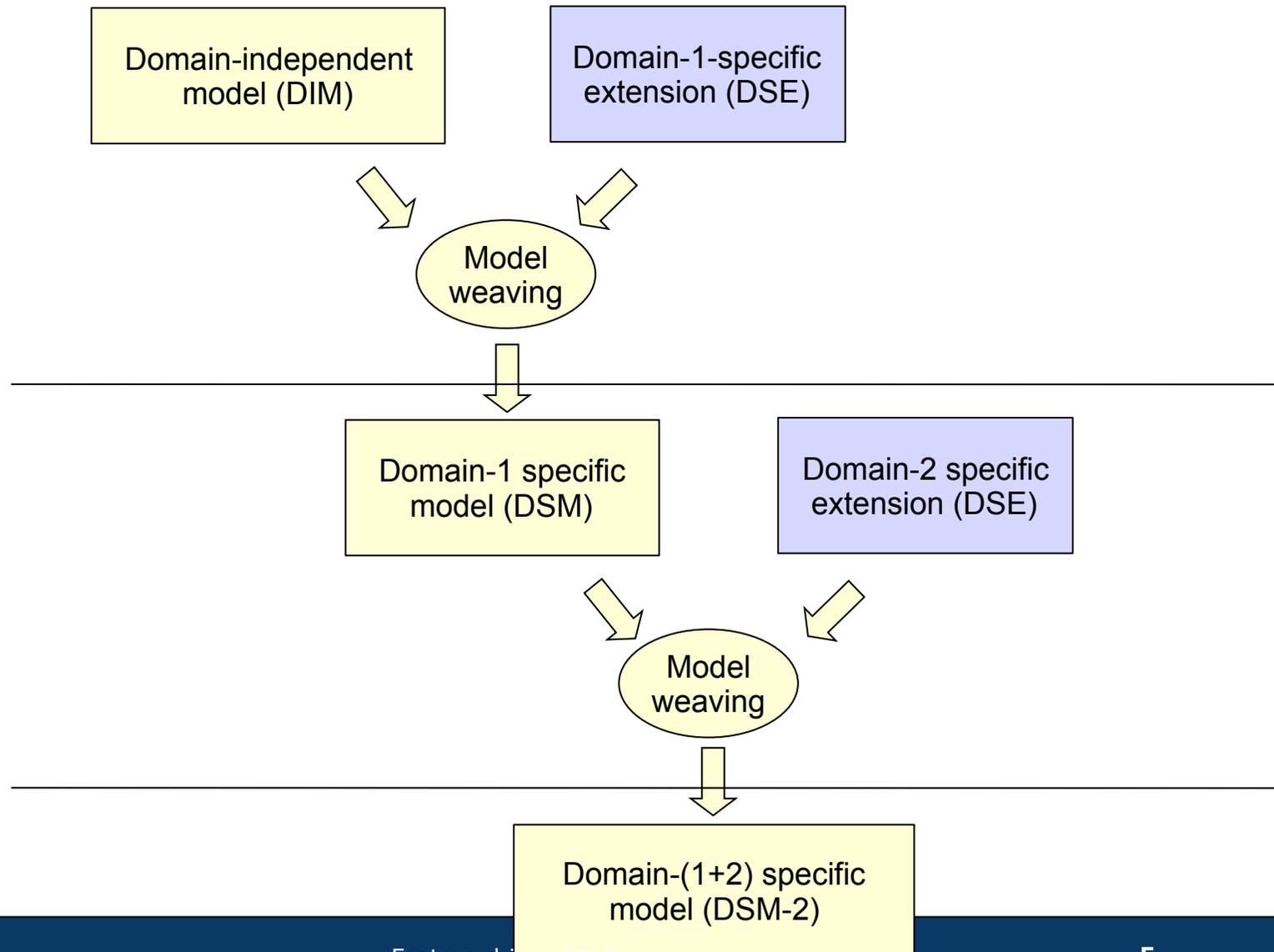**1.** Domain Models and Product Lines
**2.** Domain Ontologies and the MDA

➢ **Prof. Dr. U. Aßmann**
➢ **Florian Heidenreich**
➢ **Technische Universität Dresden**
➢ **Institut für Software- und Multimediatechnik**
➢ **Gruppe Softwaretechnologie**
➢ **http://st.inf.tu-dresden.de**
➢ **Version 12-1.0, January 23, 2013**

- Uwe Aßmann, Steffen Zschaler, and Gerd Wagner. Ontologies, meta-models, and the model-driven paradigm. In Coral Calero, Francisco Ruiz, and Mario Piattini, editors, Ontologies for Software Engineering and Technology. Springer, 2006.

- Ed Seidewitz. What models mean. IEEE Software, 20:26-32, September 2003.

requirements specification

textual requirements (stories)

Feature Model (varability model)

context model

analysis Model

domain model

use cases
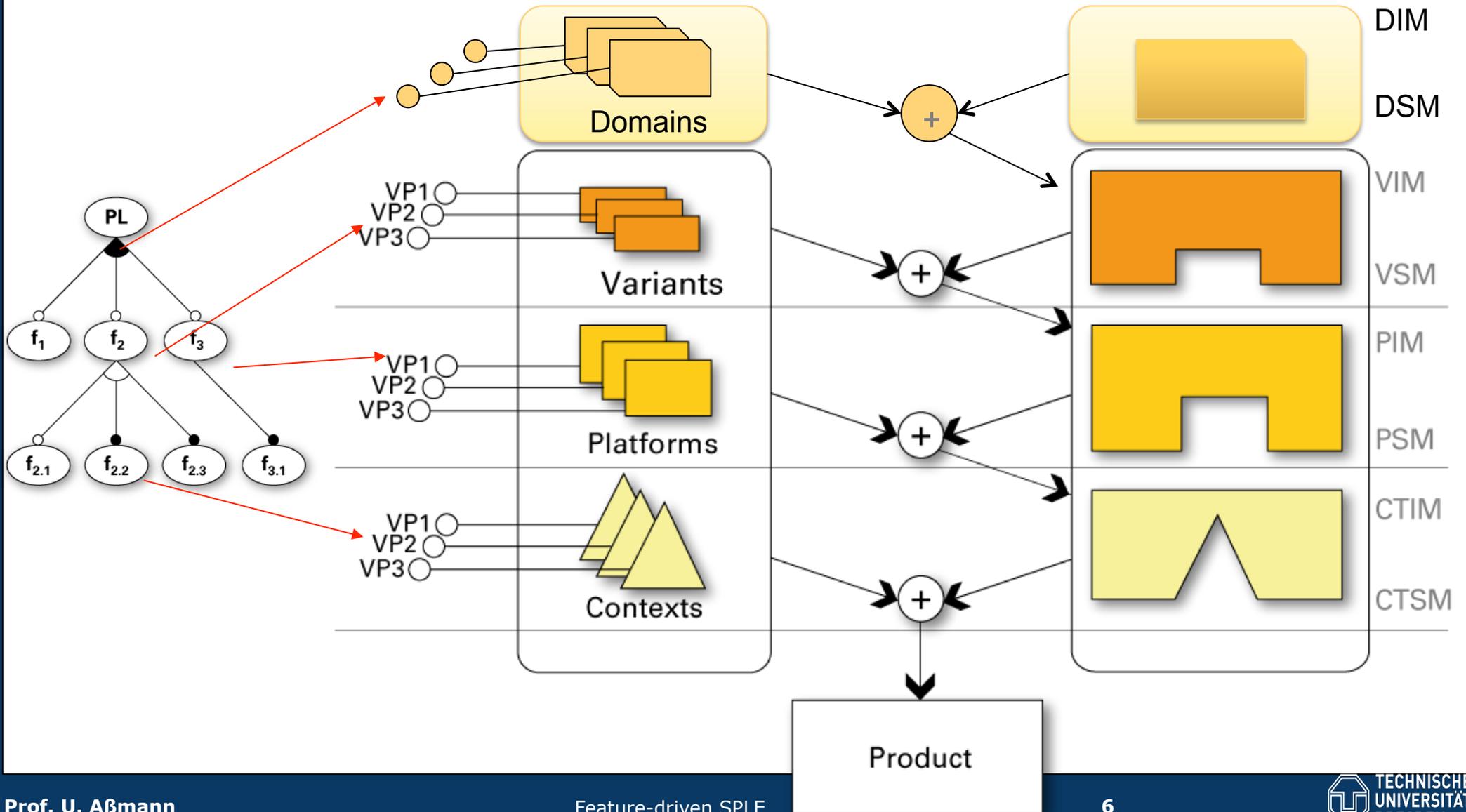
Product 1

Product 2

Product n

➢ **In a product line, domain-specific extensions can be treated like platform-specific extensions (see process FEASIPLE)**
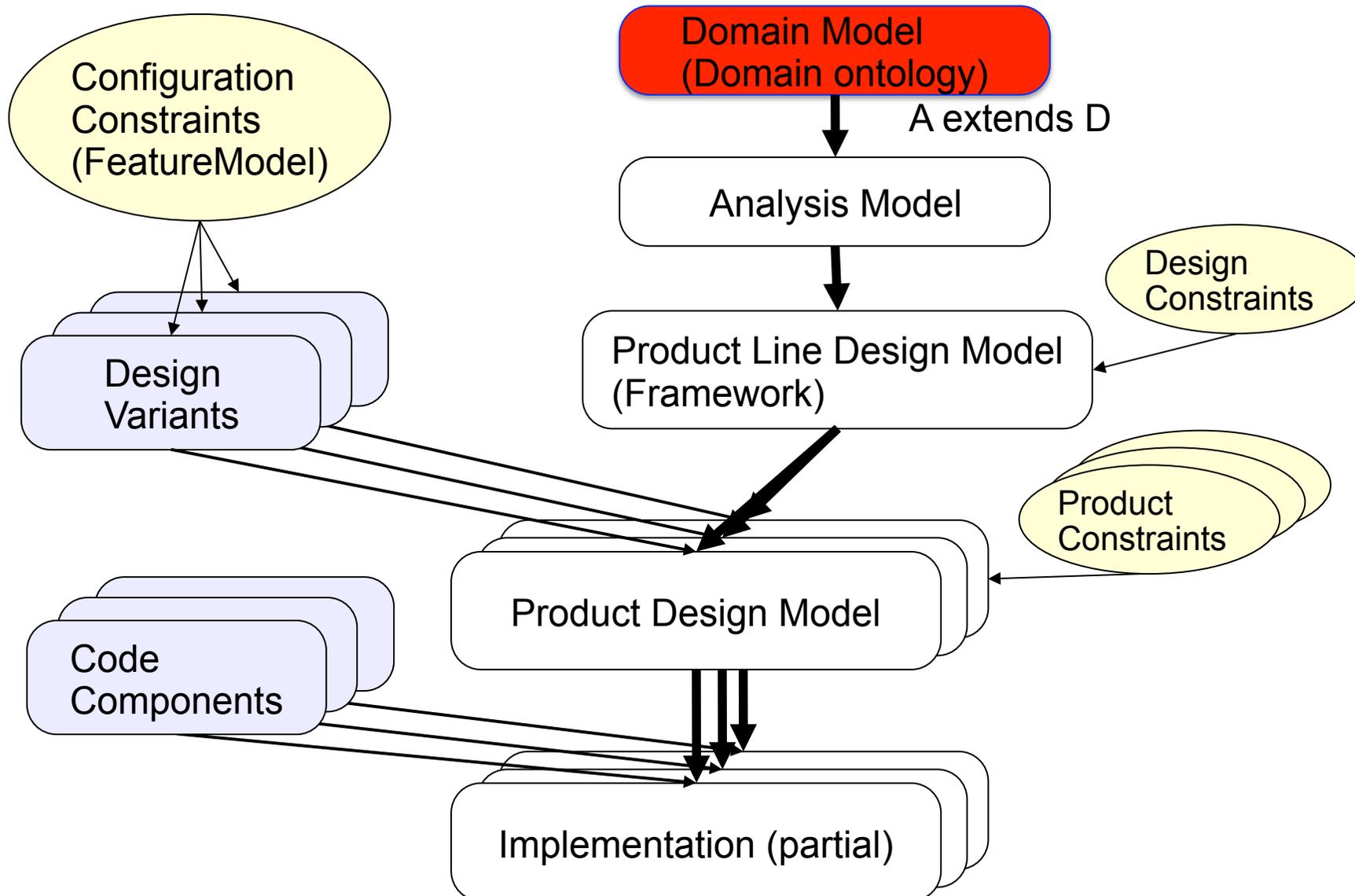
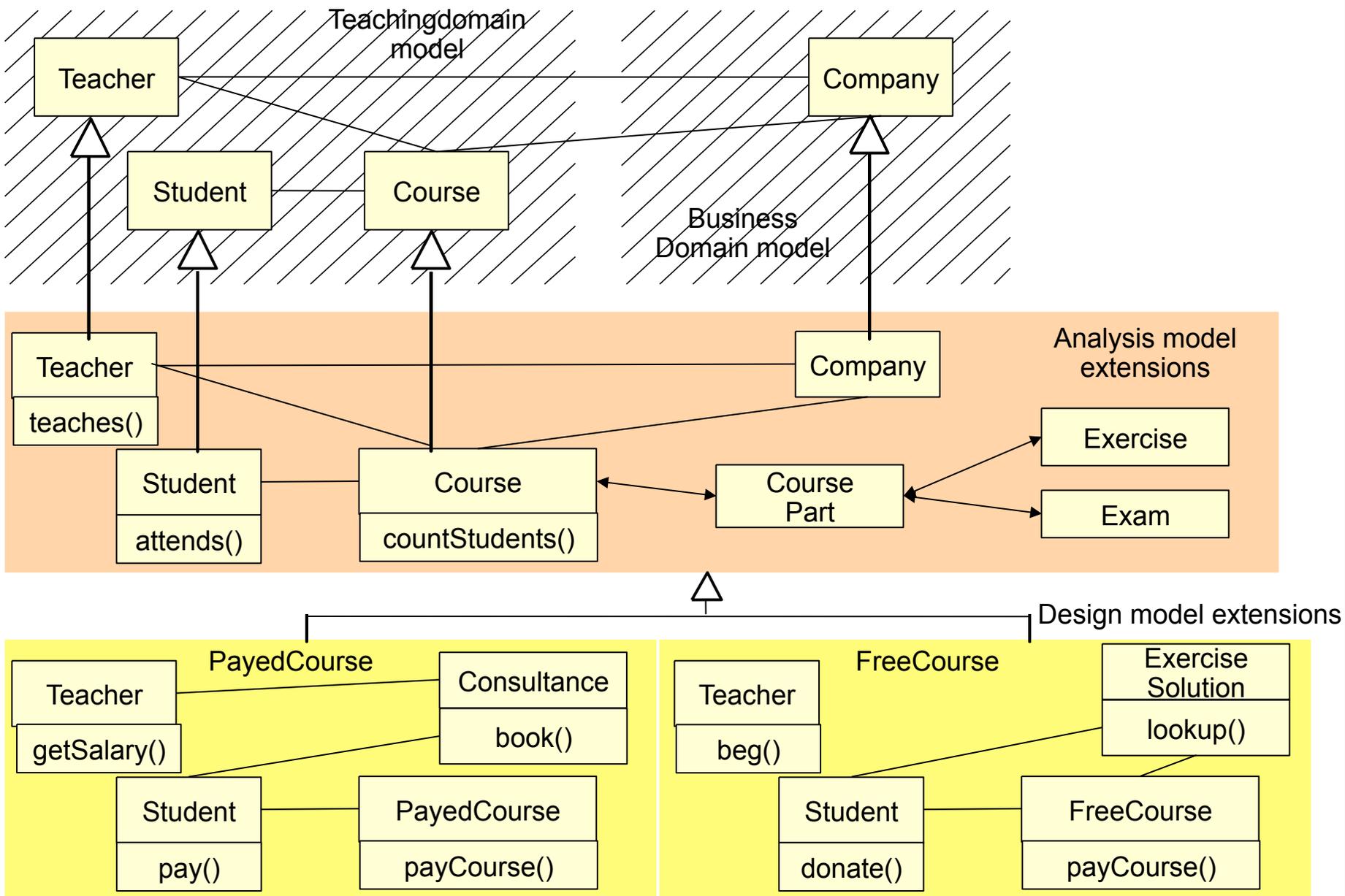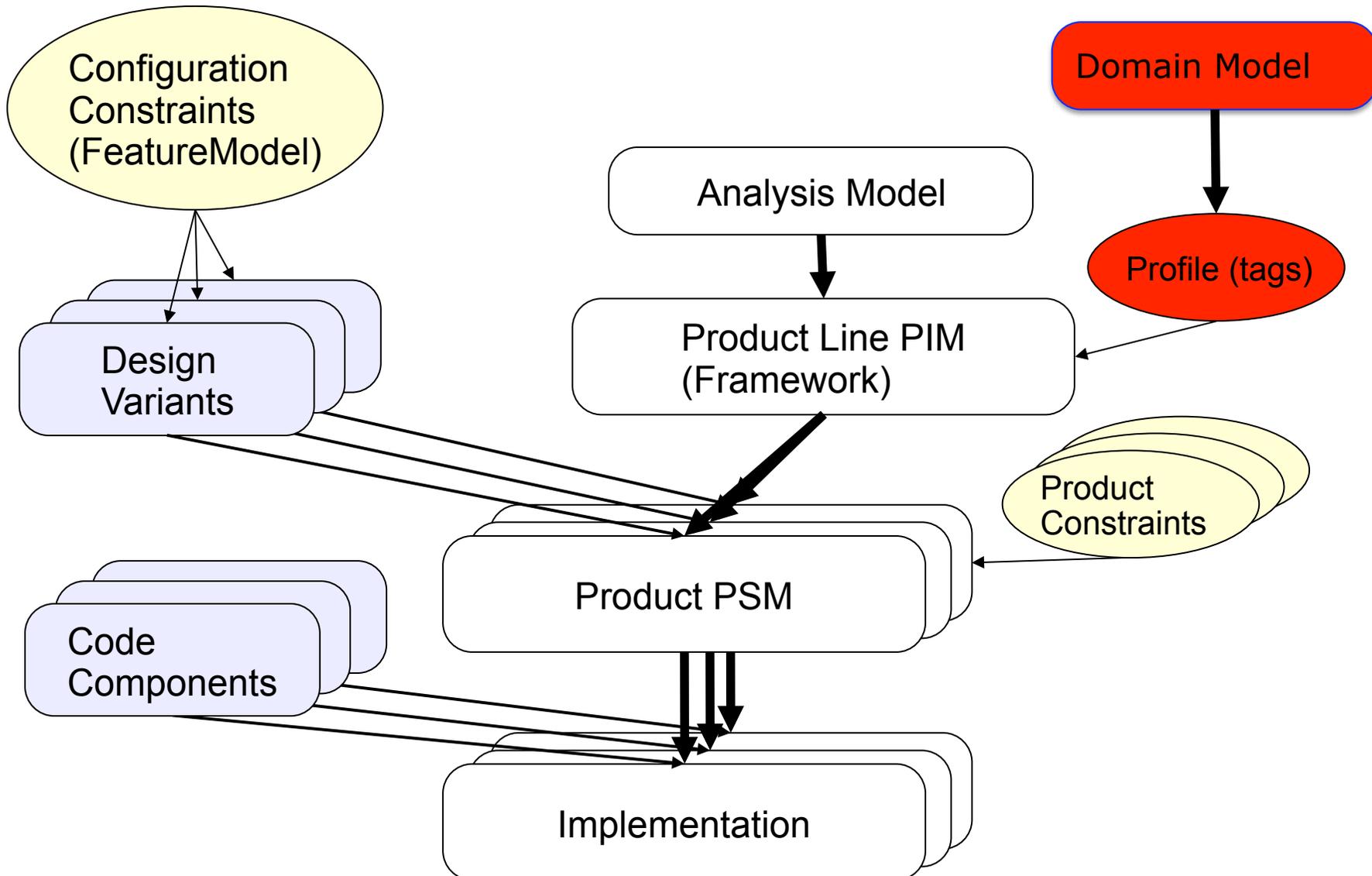> **FEASIPLE can be extended by a stage for selecting domain models**

# 32.1 DOMAIN MODELS AND SOFTWARE PRODUCT LINES (SPLC)

> As "base model" of the analysis model: analysis classes inherit from domain classes (standard was of ST-I)

➢ **Marked PIM in MDA**

➢ **Domain models are used in several places in the SPLC: As design constraints, as product constraints, as design DSL**
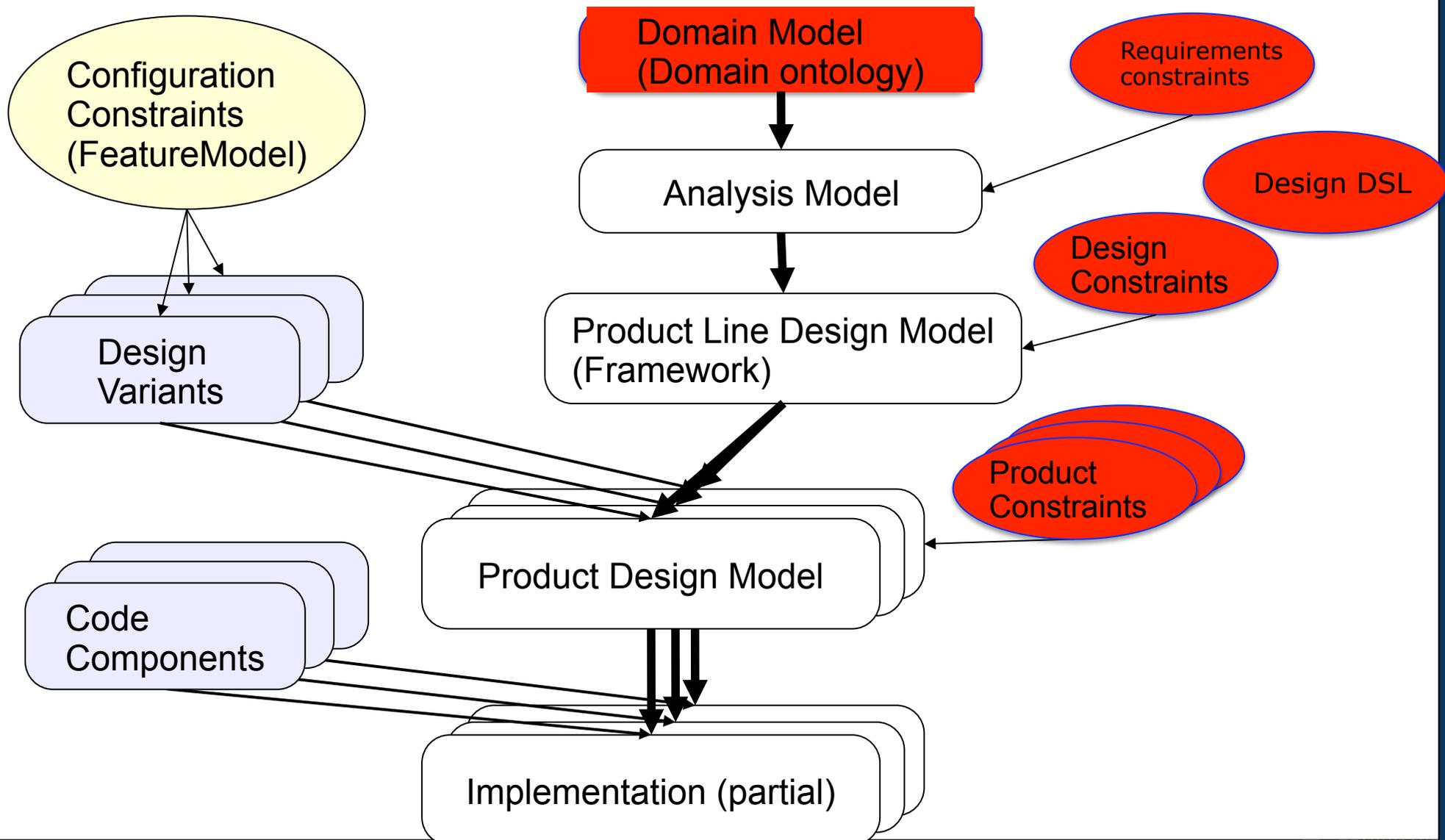
# 32.2 USING DOMAIN ONTOLOGIES IN THE MDA

- ➢ **Ontologies offer reasoning power**
  - ➢ Ontologies are modeled by domain experts and standardized
    - ➢ Gene Ontology, SnoMed, Mouse Ontology, ..
  - ➢ OWL language is standardized, reasoners are available
- ➢ **Can we use them in the Product-Line Engineering, resp. MDA?**
- ➢ **How do ontologies and system models relate?**
  - ➢ Ontology
  - ➢ Metamodels
  - ➢ Model-Driven Engineering (MDE)
  - ➢ Model-Driven Architecture (MDA)

➢ **How can we find a place for ontologies in the world of MDA?**

A model is an external and explicit representation of a part of reality as seen by the people who wish to use that model to understand, change, manage, and control that part of reality.  [Pidd]

A model of a system is a description or specifiation of that system and its environment for some certain purpose. [MDA Guide]

But....

Ontologies are formal explicit specifications of a shared conceptualization.[Gruber]

TECHNISCHE
UNIVERSITÄT
DRESDEN

➢ **[Aßmann, Zschaler, Wagner 06]**

An **ontology:**

a standardized,
descriptive model,

representing reality
by a set of concepts, their
interrelations, and constraints
under
open-world assumption.

A **system model:**

a non-standardized,
prescriptive model,

representing a set of systems
by a set of concepts, their interrelations,
and constraints
under
closed-world assumption.

A model can be *descriptive* or *prescriptive*.
[Seidewitz CACM 03]

- Models describe or control reality.
- If they describe, they monitor reality and form true, or faithful, abstractions (Analysis, Reengineering)
- If they control, they prescribe reality (Construction, Specification)

► Ontologies need the **open-world assumption**

- Analysis perspective
- **Anything not explicitly expressed is unknown**
- Ontologies use a form of partial description to abstract

► System models need **closed-world assumption**

- Design perspective
- **Anything not explicitly expressed is wrong**
- System models specify completely

Descriptive

Prescriptive

➤ **With Closed World Assumption (Reasoning)**
  ➤ Querying
    ➤ needs CWA to exclude erroneous data
  ➤ Metamodeling:
    ➤ needs CWA to exclude erroneous programs
  ➤ Integrity constraints
    ➤ needs CWA to exclude erroneous models

➤ **With Open World Assumption**
  ➤ Domain modeling
    ➤ needs OWA because of partial specification of domain

➢ **A *technical space* uses a metapyramid, formed by a specific metalanguage on M3**

➢ **More in course "Softwarewerkzeuge" (WS)**

| Level | Modelling (left) | | MOF (right) |
|---|---|---|---|
| **M3 metametamodel Level (metalanguage)** | Modelling concepts | | MOF |
| | ↑ validInstanceOf | | ↓ describes |
| **M2 metamodel Leve (language)l** | Language descriptions | | UML, CWM, ER, OWL |
| | ↑ validInstanceOf | | ↓ describes |
| **M1 model and program level** | Types, programs, models domain ontologies | | |
| | ↑ validInstanceOf | | ↓ describes |
| **M0 run-time object level** | Software objects describing world objects | | model instances |

TECHNISCHE UNIVERSITÄT DRESDEN

M3 metametamodel level

MetaMetamodels

M2 metamodel level

Metamodels (languages)

M1 model level

Models

M0 object level

Modelling concepts

MOF

CIM-MM

Language specifications

PIM-MM

<<instance-of>>

PSM-MM

Instance specifications

CIM

<<instance-of>>

PIM

PSM

System instances

<<instance-of>>

TECHNISCHE UNIVERSITÄT DRESDEN

Analysis
Problem Domain
Descriptive

Design
Solution Domain
Prescriptive (Specifications)

M3 metametamodel level

**Universal Metalanguage
With Closed-World Assumption or
Open-World Assumption**

M2 metamodel level

<<instance-of>>

<<instance-of>>

Upper ontologies

Metamodels
(languages, language concepts)

M1 model level

<<is-a>>

<<instance-of>>

Domain ontologies

Models

M0 object level

<<described-by>>

<<instance-of>>

Real world objects

Software objects

TECHNISCHE
UNIVERSITÄT
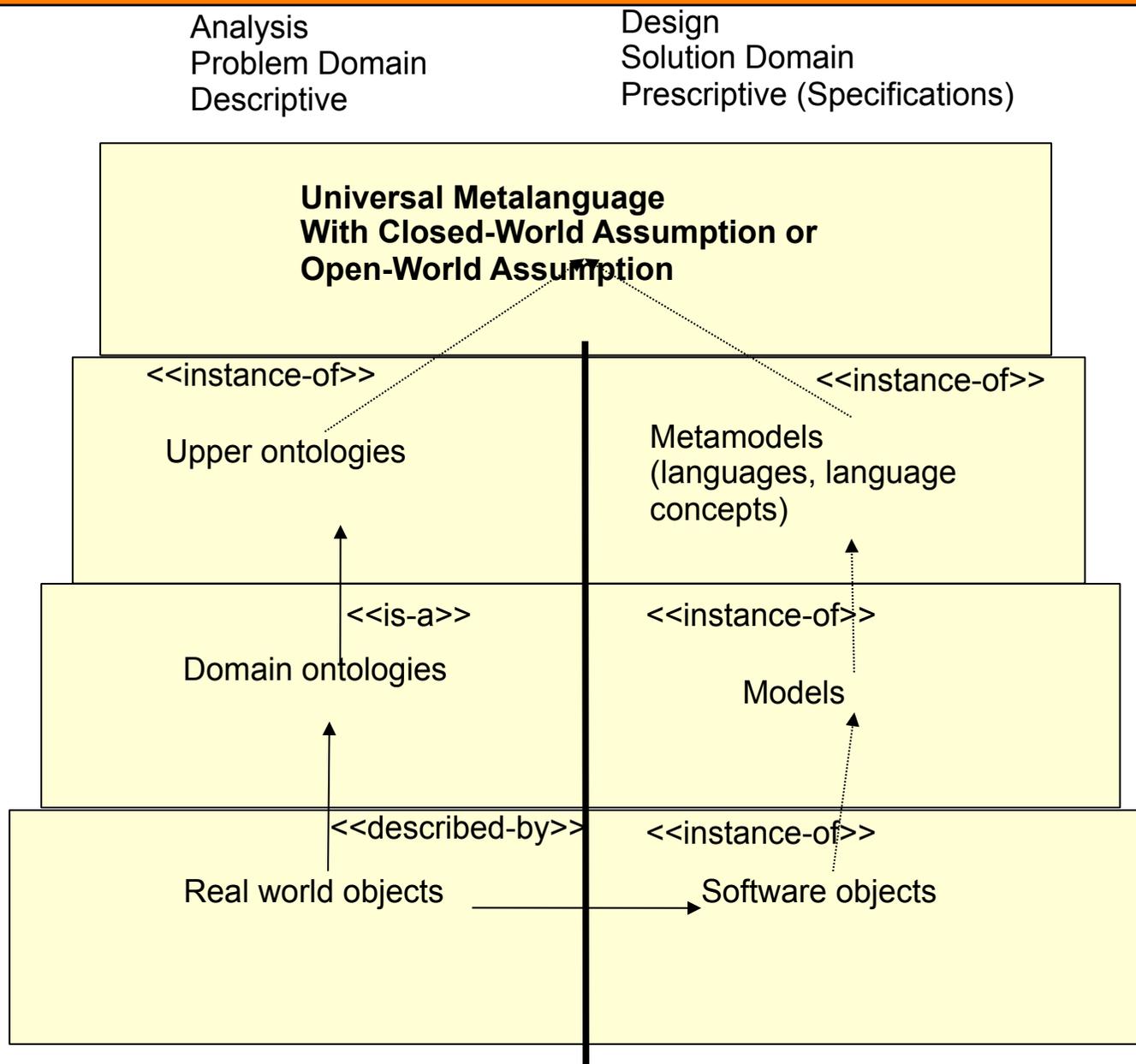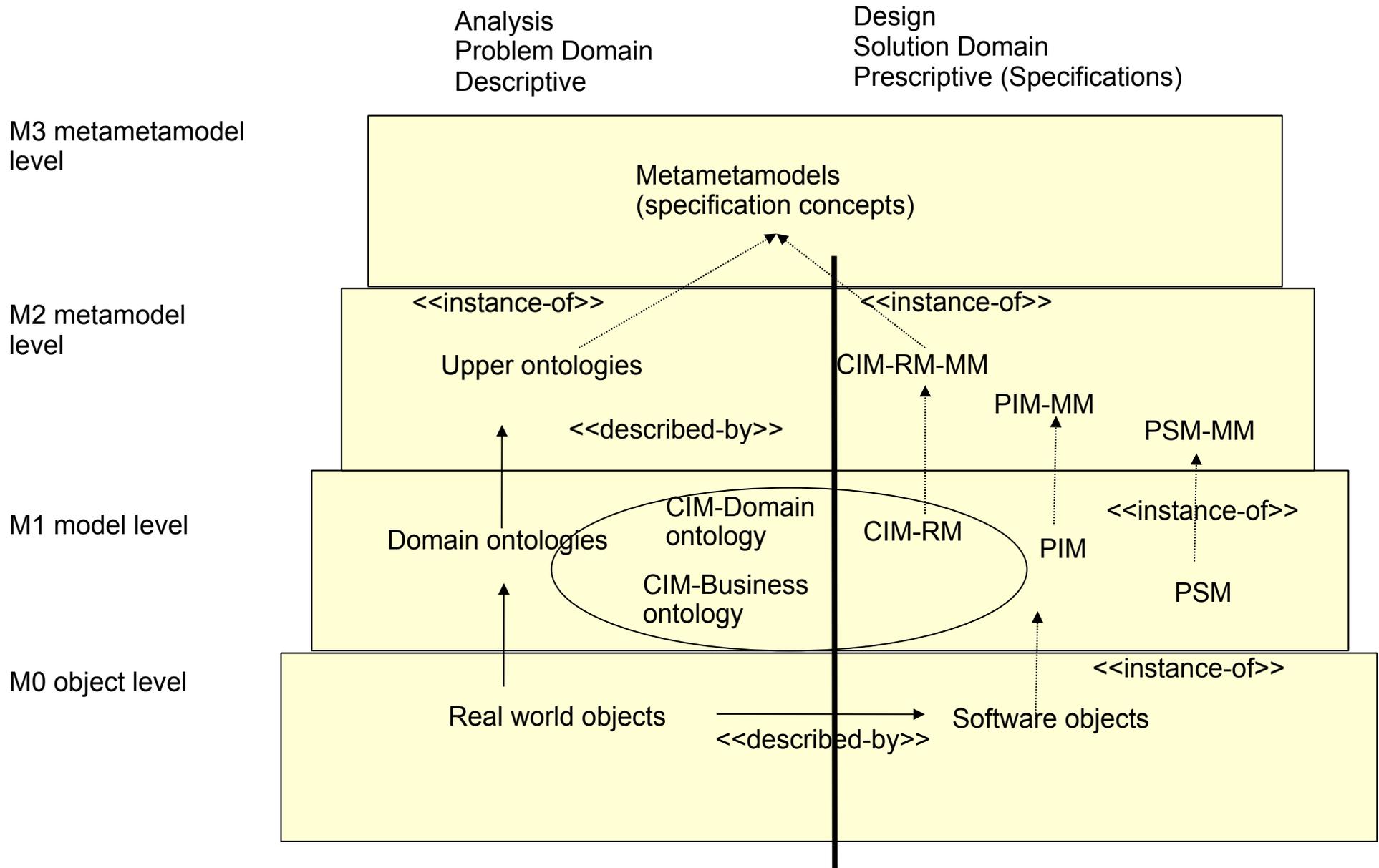DRESDEN

- ➢ **Ontologies are advantageous in SPLE for**
  - ➢ domain ontologies
  - ➢ integrity constraint ontologies in product lines
- ➢ **but...**
  - ➢ Ontologies should not be misused as system models
  - ➢ Ontologies *complement* system models
  - ➢ Ontologies in OWA for domain modeling, CWA for the rest
- ➢ **Integration technology and tools needed!**
- ➢ **MOST project (Marrying Ontologies and Software Technology)**
- ➢ **www.most-project.eu**

**Softwaretechnologie II**
Modellierung Entwurfsmethoden
Elementares über Produktlinien

**Design Patterns and Frameworks (WS)**
Produktlinien mit objektorientierter Programmierung (OO Komponentenmodell)

**Component-Based Software Engineering (SS)**
Produktlinien mit anderen Komponentenmodellen

**Ausgewählte Kapitel der Softwarearchitektur (SS)**
(Richly)
Ontologies, Method engineering, IT landscapes

**Software-Werkzeuge (WS)**
Produktlinien mit Werkzeugen
Technikräume

**Automotive Software Engineering (SS)**
(Prof. Hohlfeld)

**Academic Skills for Software Engineers (WS)**
Wie man wissenschaftlich arbeitet

**Software-Management (SS)**
Wie man Projekte macht

**Enterprise Software (WS)**
(Prof. Kubach)