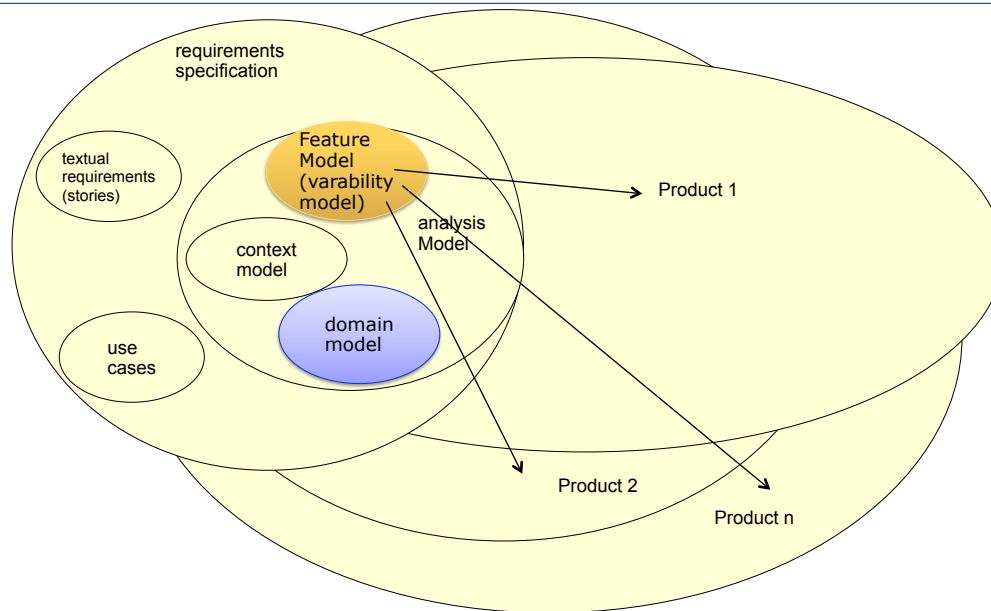# Slide 1

32) Domain Models and Product Lines

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 11-0.1, Dezember 28, 2011**

Slide 2

➢ **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.**

Slide 3

So Far: Product Lines Configured by Feature Models

Slide 4

Now: Product Lines with different Domain Models

Slide 5
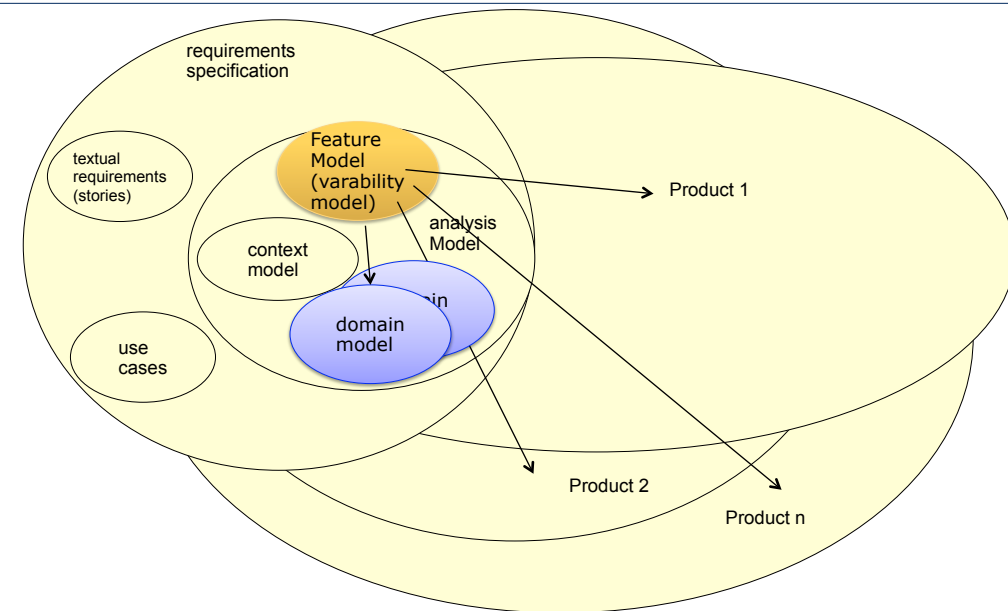
➢ **In a product line, domain-specific extensions can be treated like platform-specific extensions (see process FEASIPLE)**

Slide 6
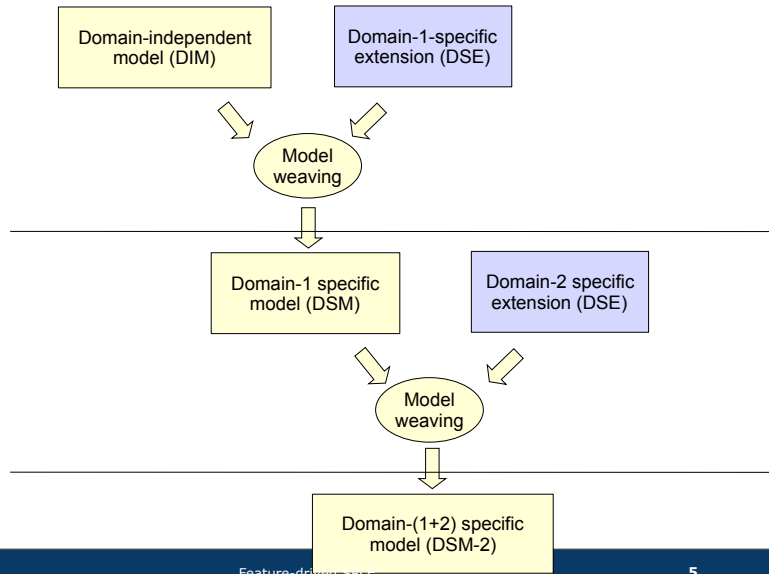
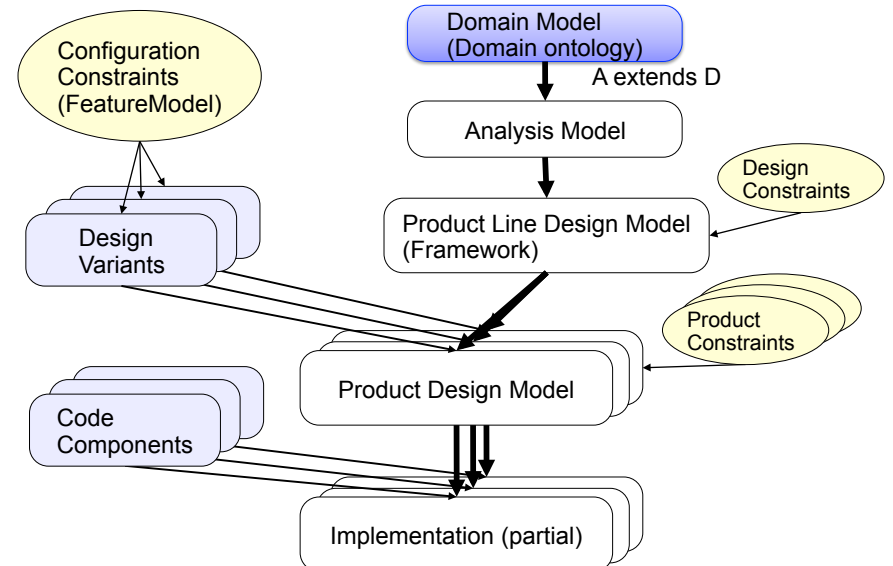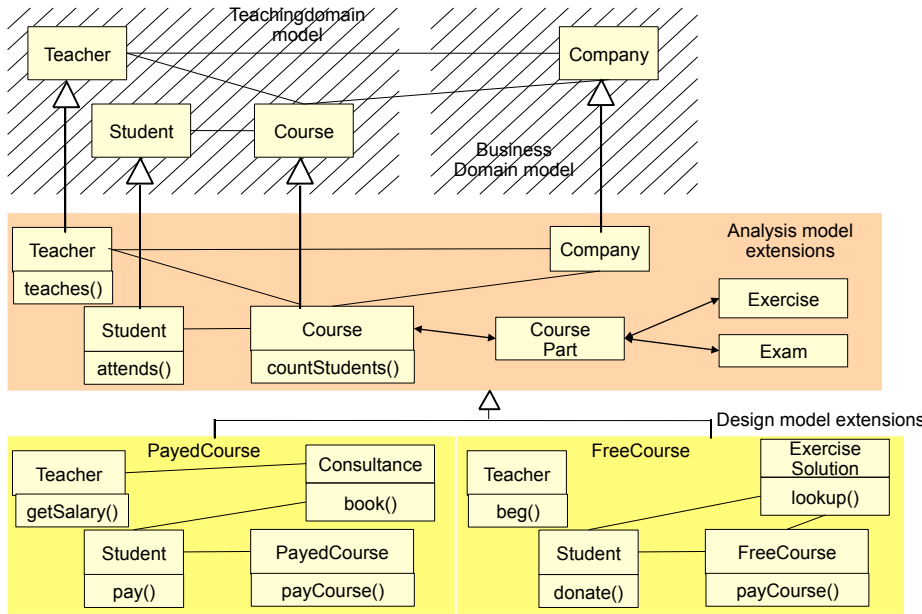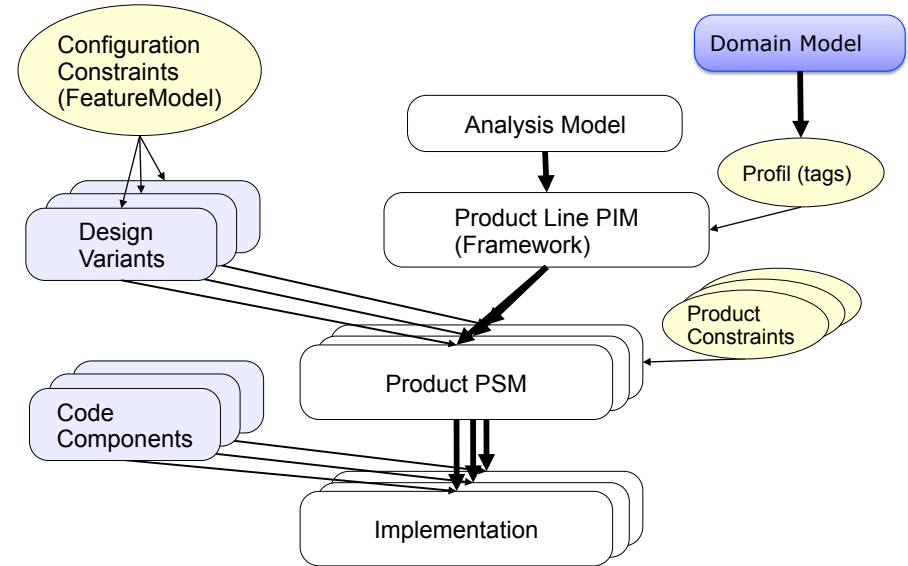➢ **FEASIPLE can be extended by a stage for selecting domain models**

Slide 7
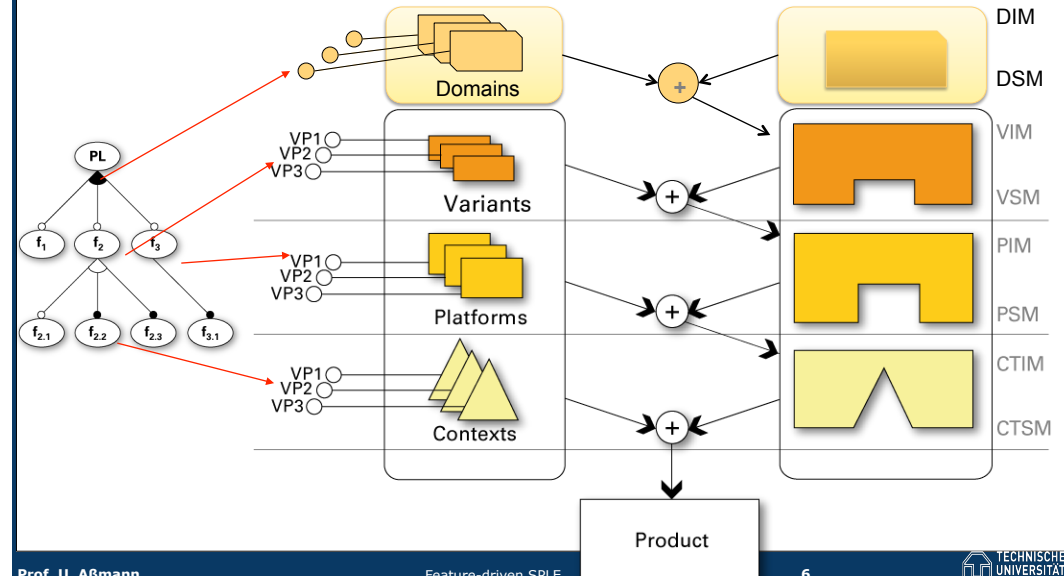
32.1 DOMAIN MODELS AND PRODUCT LINES

Slide 8

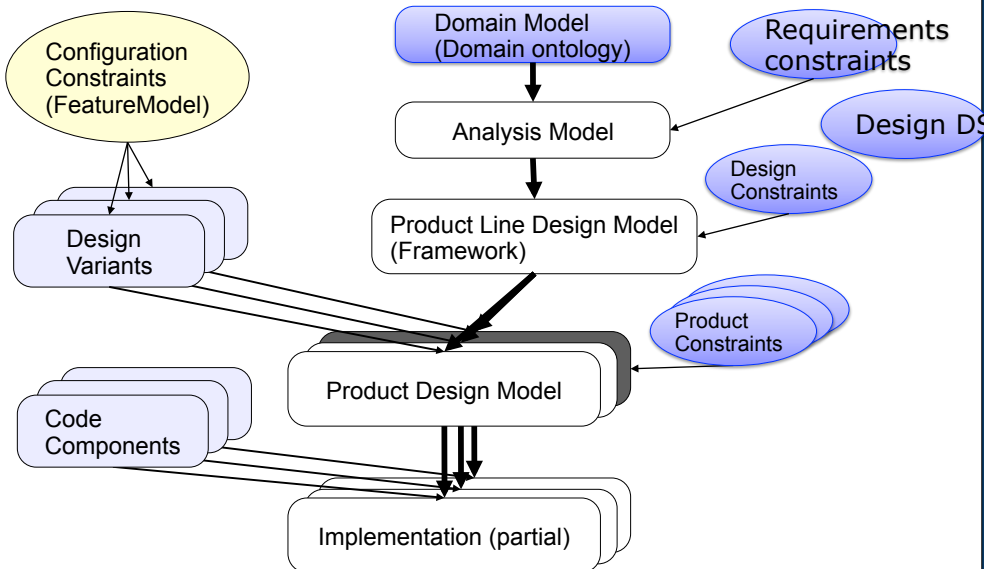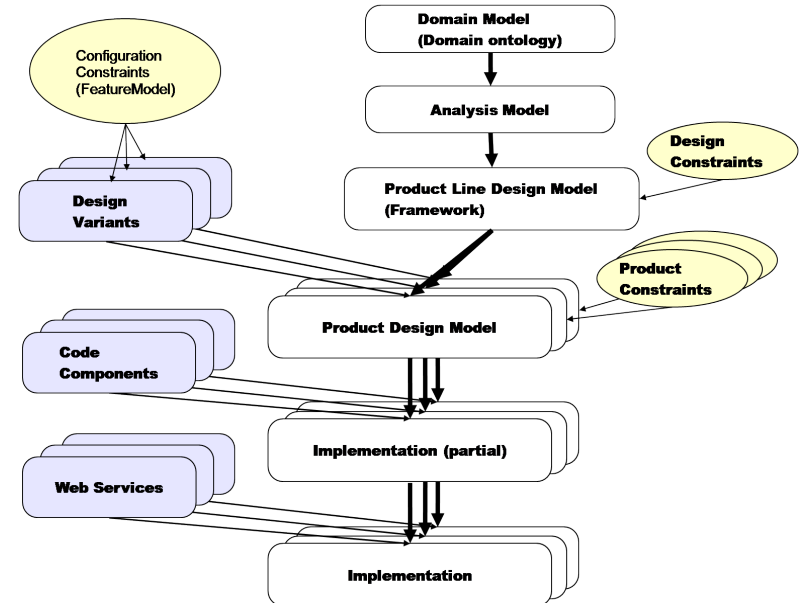➢ **As supermodel of the analysis model: analysis classes inherit from domain classes (standard was of ST-I)**

Slide 9 — Domain Models as Base Models

Teachingdomain model

- Teacher
- Student — Course
- Company

Business Domain model

Analysis model extensions
- Teacher — teaches()
- Student — attends()
- Course — countStudents()
- Company
- Course Part
- Exercise
- Exam

Design model extensions

PayedCourse
- Teacher — getSalary()
- Consultance — book()
- Student — pay()
- PayedCourse — payCourse()

FreeCourse
- Teacher — beg()
- Student — donate()
- FreeCourse — payCourse()
- Exercise Solution — lookup()

Slide 10 — Marking with Tags of Domain Profiles

➢ **Marked PIM in MDA**

- Configuration Constraints (FeatureModel)
- Design Variants
- Code Components
- Domain Model
- Analysis Model
- Profil (tags)
- Product Line PIM (Framework)
- Product Constraints
- Product PSM
- Implementation

Slide 11 — MOST Product Family Architecture (MOPF) As Contraints

➢ **As design constraints, as product constraints, as domainenmodell**

- Configuration Constraints (FeatureModel)
- Domain Model (Domain ontology)
- Requirements constraints
- Analysis Model
- Design DS
- Design Variants
- Design Constraints
- Product Line Design Model (Framework)
- Product Constraints
- Code Components
- Product Design Model
- Implementation (partial)

Slide 12

- Configuration Constraints (FeatureModel)
- Domain Model (Domain ontology)
- Analysis Model
- Design Variants
- Design Constraints
- Product Line Design Model (Framework)
- Product Constraints
- Code Components
- Product Design Model
- Web Services
- Implementation (partial)
- Implementation

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]

➢ **[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**
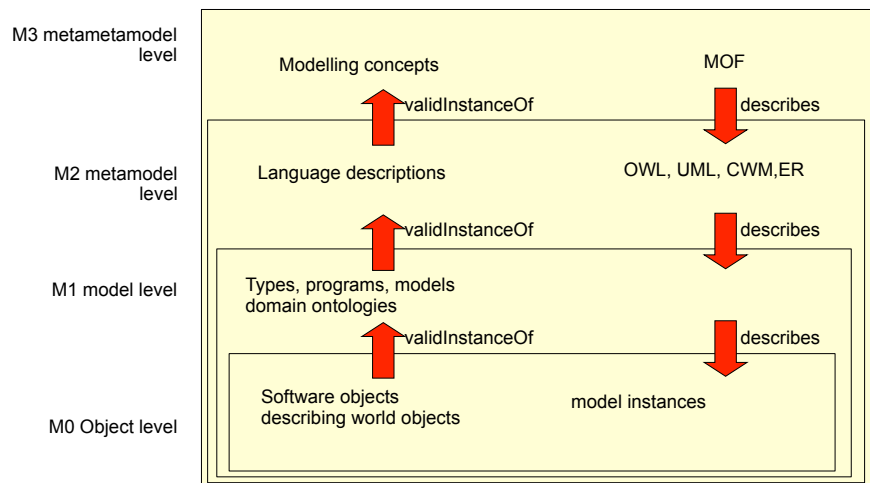 - Ontologies use a form of partial description to abstract

► System models need **closed-world assumption**
  - Design perspective
  - **Anything not explicitly expressed is wrong**
  - System models specify completely

Descriptive

Prescriptive

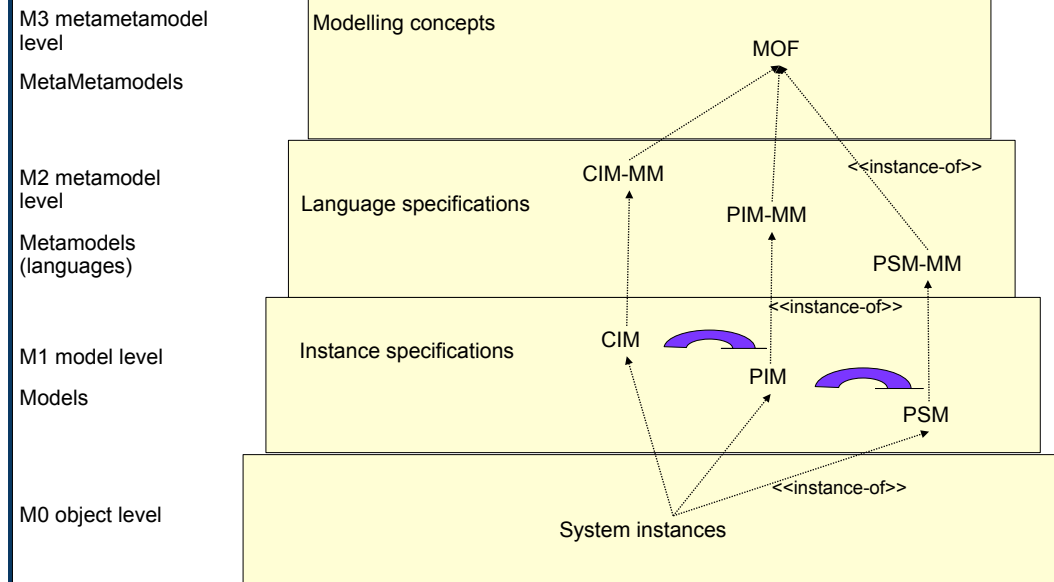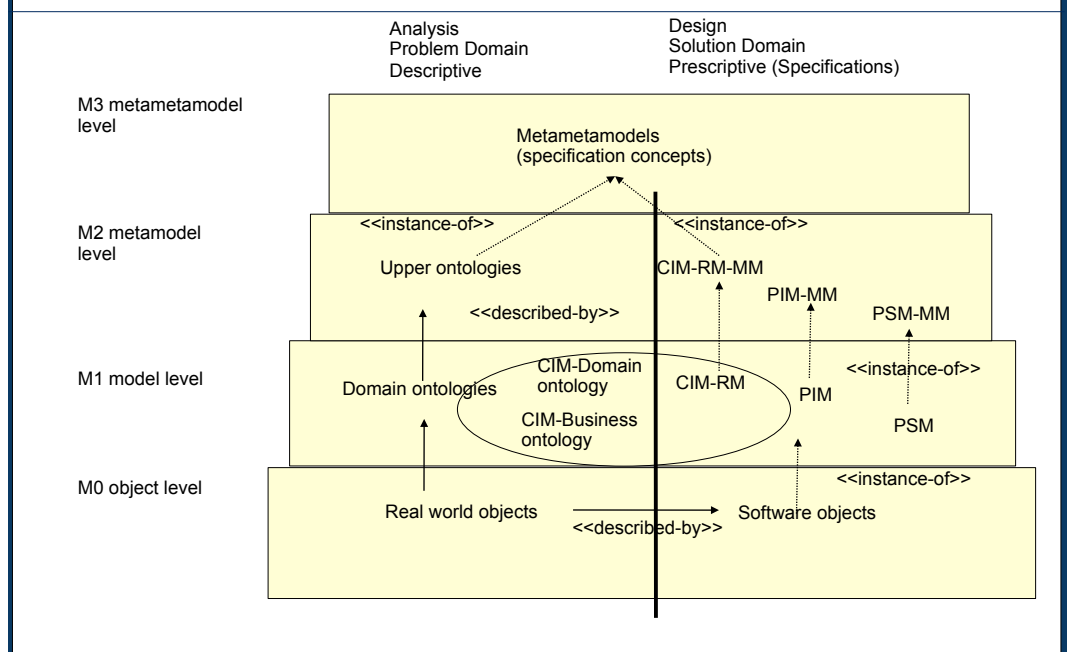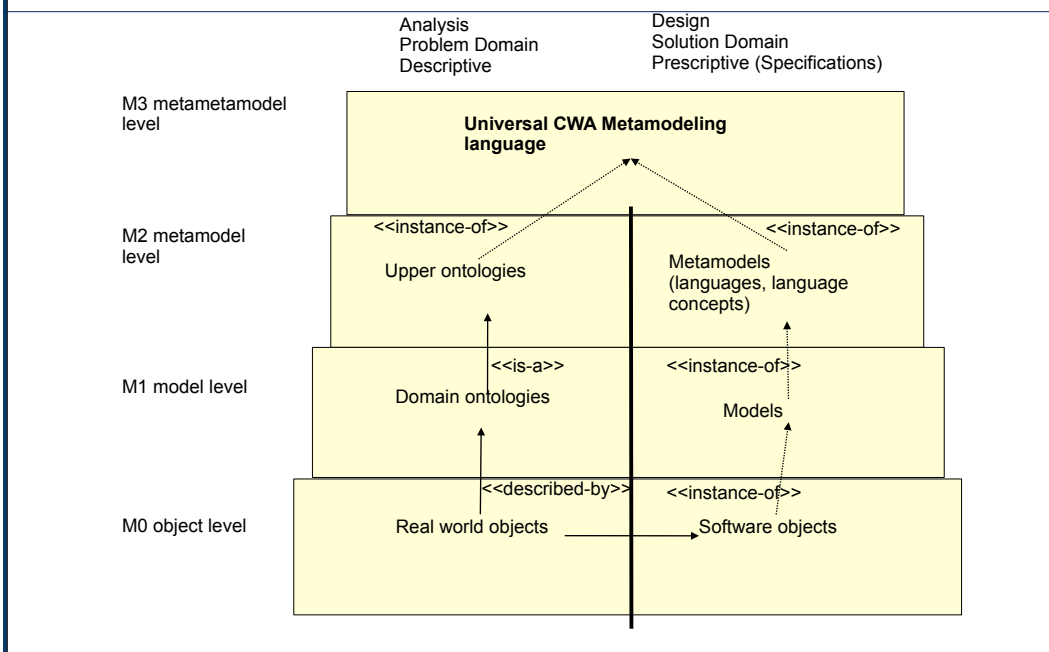
➤ **With Closed World Assumption (Reasoning)**
  ➤ Querying
    ➤ needs CWA to exclude erroneous data
  ➤ Metamodeling:
    ➤ needs CWA to exclude erroneous programs
  ➤ Integrity constraints
    ➤ needs CWA to exclude erroneous models
➤ **With Open World Assumption**
  ➤ Domain modeling
    ➤ needs OWA because of partial specification of domain

➤ **aka *metapyramid***

Analysis
Problem Domain
Descriptive

Design
Solution Domain
Prescriptive (Specifications)

M3 metametamodel level — **Universal CWA Metamodeling language**

M2 metamodel level — <<instance-of>> — Upper ontologies — <<instance-of>> — Metamodels (languages, language concepts)

M1 model level — <<is-a>> — Domain ontologies — <<instance-of>> — Models

M0 object level — <<described-by>> — Real world objects — <<instance-of>> — Software objects

---

Analysis
Problem Domain
Descriptive

Design
Solution Domain
Prescriptive (Specifications)

M3 metametamodel level — Metametamodels (specification concepts)

M2 metamodel level — <<instance-of>> — Upper ontologies — <<described-by>> — <<instance-of>> — CIM-RM-MM — PIM-MM — PSM-MM

M1 model level — Domain ontologies — CIM-Domain ontology — CIM-Business ontology — CIM-RM — PIM — <<instance-of>> — PSM

M0 object level — Real world objects — <<described-by>> — <<instance-of>> — Software objects

---

Conclusions

➢ **Ontologies are advantageous in PLE 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**

Master`s Courses
(weiteres Hauptstudium)

Softwaretechnologie II
Entwurfsmethoden
Elementares über Produktlinien

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

Software-Werkzeuge
Produktlinien mit Werkzeugen
Technikräume

Component-Based Software Engineering
Produktlinien mit anderen Komponentenmodellen

Software-Management
Wie man Projekte macht