

3. Modeling (Part I)

Engineering Web and Data-intensive Systems

Dr. Volker Riediger - Winter Term 2022/23

Modeling (Part I)

- Purpose of Models
- Types of Models
- Model Driven Engineering
- Basics of the UWE approach



Image: colourbox.de

Models

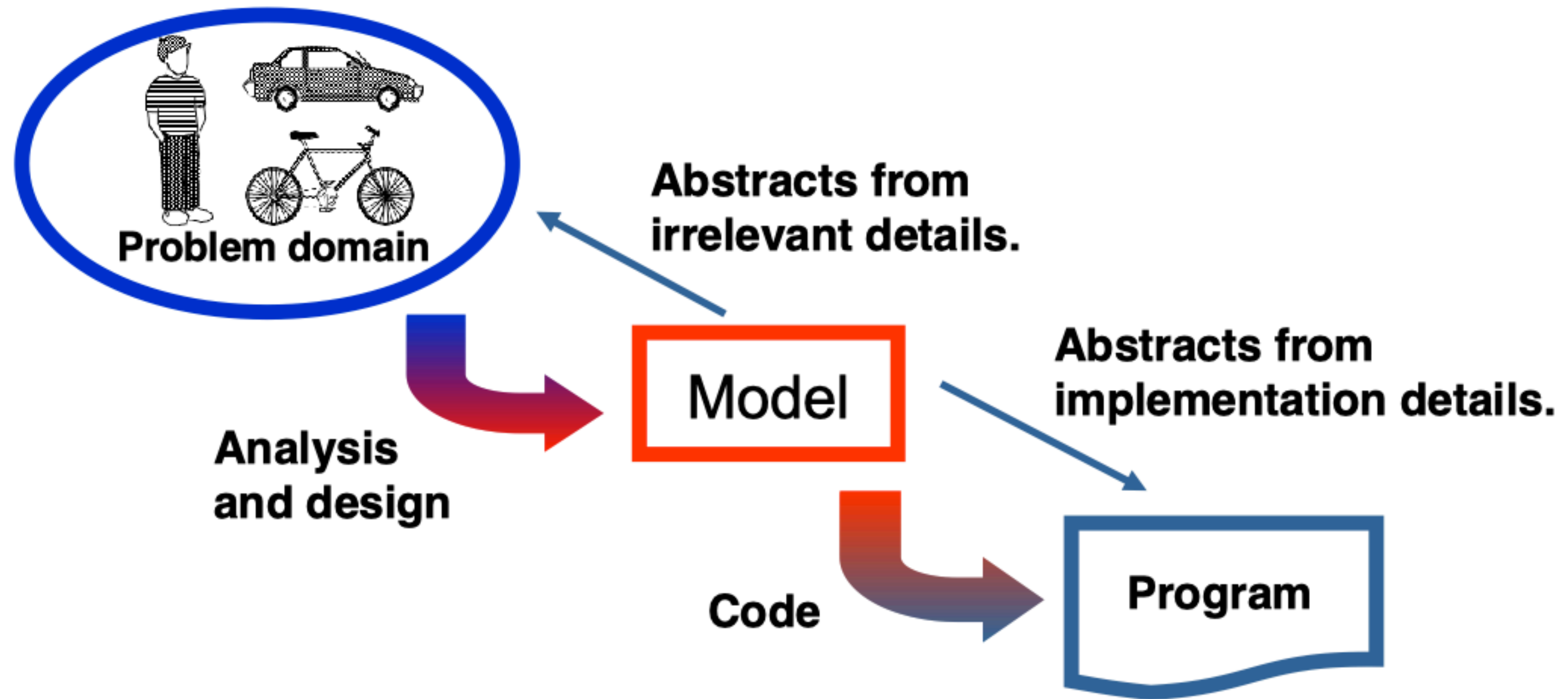
- A **model** is a **purpose-specific abstraction** of (an aspect) of the subject area.
- Abstraction enables modelers to **simplify** the problem by **omitting** parts that are not relevant.

- Models can be expressed in a multitude of representations
 - Text
 - Diagrams
 - Mathematics
 - Domain-specific notations
 - ...
- Models are used in virtually all phases of the software life cycle.

Models in SE

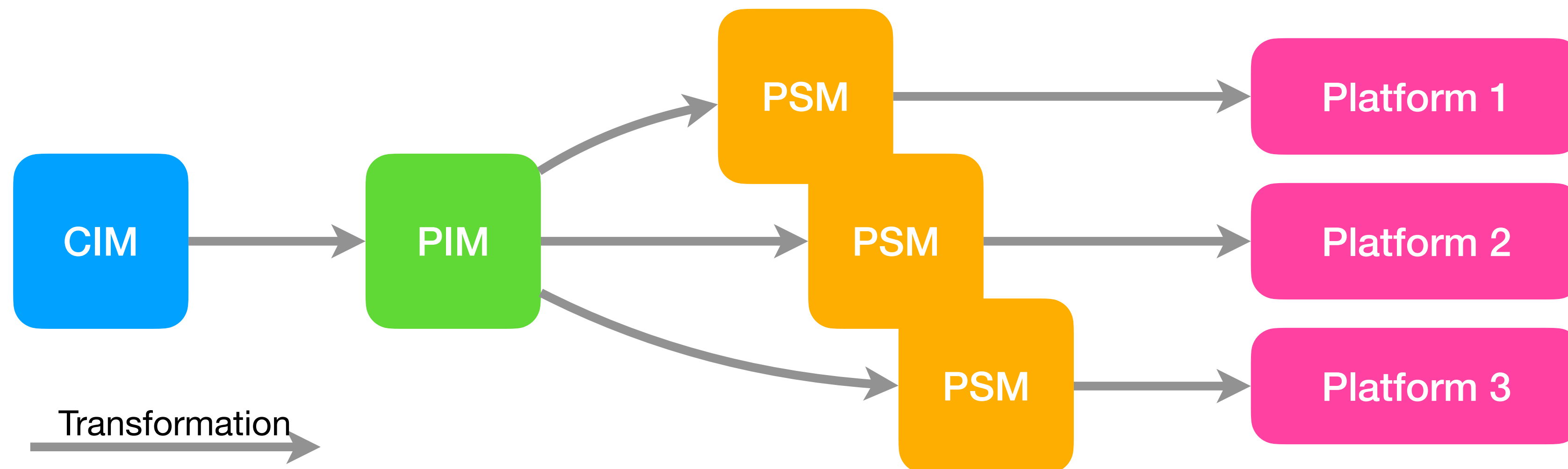
- Models in Software Engineering serve many **purposes**
 - **Reducing the complexity** of real-world problems
 - **Communication** with stakeholders
 - **Documentation** (descriptive use)
 - **Specification** (prescriptive use)
 - Source and/or target of **transformations**
- Depending on the purpose, models have
 - different **granularity** or level of detail
 - different **degree of formalization**
 - **precisely defined semantics** required for automatic transformations

Model Based Software Engineering

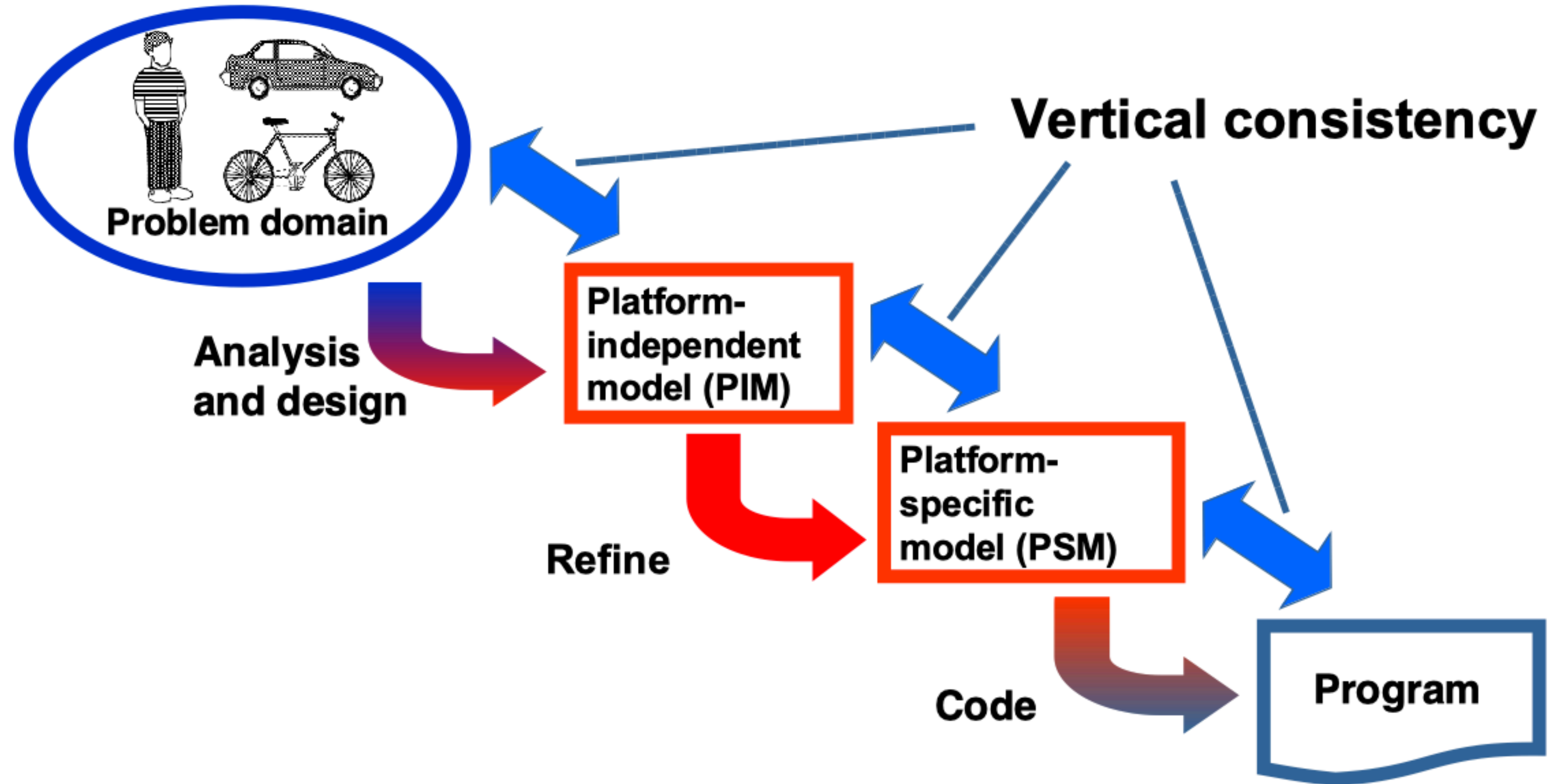


MDE / MDA

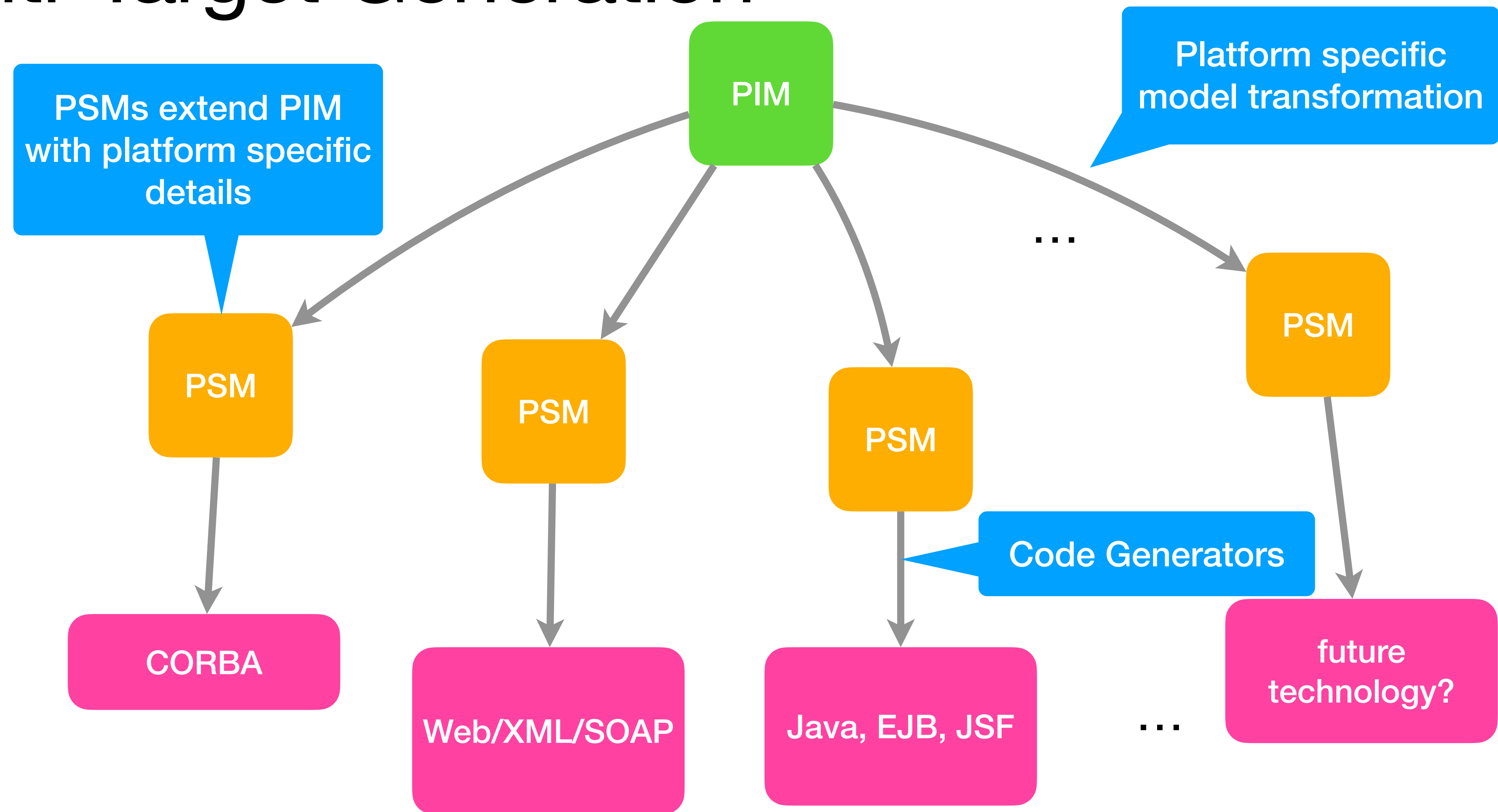
- In Model Driven Engineering (MDE) and Model Driven Architecture (MDA), models play a central role.
- MDE employs (partly automatic) **transformations** to realize a software system
- CIM - Computation Independent Model
- PIM - Platform Independent Model
- PSM - Platform Specific Model



MDA



Multi-Target Generation



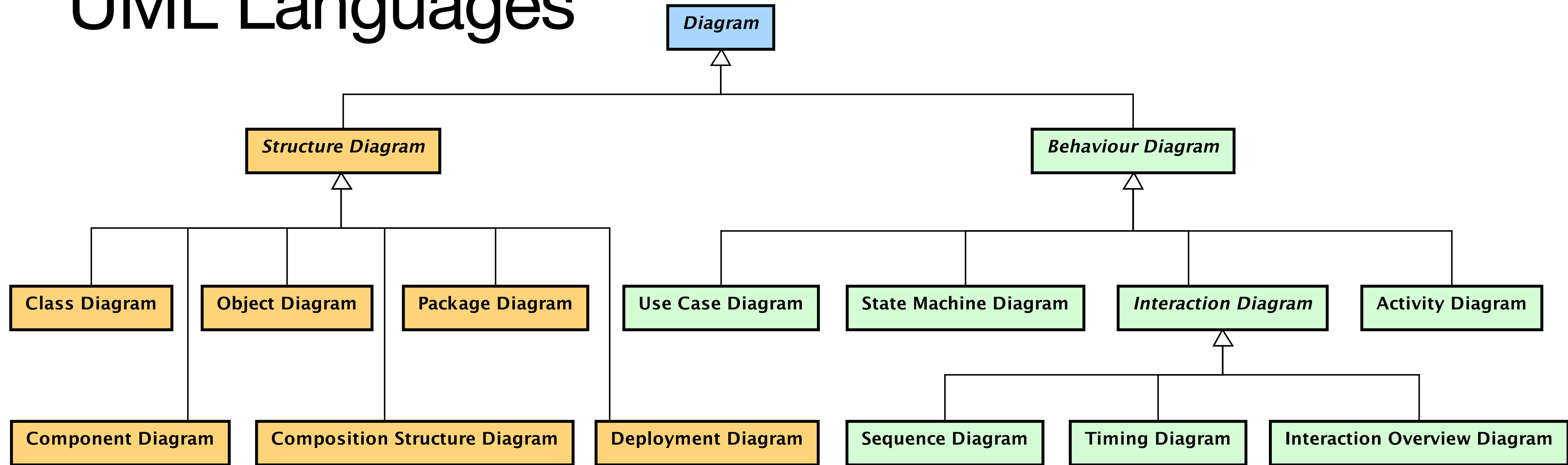
Modeling Languages

- Models can be expressed in a multitude of languages
 - Textual
 - Graphical, diagrams
 - Mathematical
 - Domain-specific notations
 - ...
- Modeling languages come with:
 - **Syntax** - What are the **elements**, **rules**, and **constraints** for valid models?
Syntax is often defined by meta models.
 - **Semantics** - What is the **meaning** of the models?
Formulated in a range of notations from natural language to mathematical calculus
 - **Pragmatics** - Who can the language **be used**?
Best practice advice, tutorials, templates, examples, ...

UML - Unified Modeling Language

- UML is a family of visual languages for the description of software systems.
- UML is developed by the the Object Management Group (OMG), an organization of more than 800 members from industry, users, and research.
- „Three Amigos“ G. Booch, I. Jacobson, J. Rumbaugh
- UML is evolving constantly
 - 1995: UML 0.8
 - 1996: UML 0.9
 - 1997: UML 1.0
 - ...
 - 2005: UML 2.0
 - ...
 - 2011: UML 2.4.1
(also adopted by ISO)
 - 2015: UML 2.5
 - 2017: UML 2.5.1
 - <https://www.omg.org/spec/UML/2.5.1>

UML Languages



- Diagram languages for **structural** and **behavioral** models
- Customization and **extension** by UML **profiles**
- Many derivatives, e.g. BPMN, SysML, ...

Models for Structure

- **Coarse** grained (high level, user perspective, requirements)
 - **Domain**/Information model, packages
business objects and their and relations
implementation details
 - Example **instances**
concrete objects to explain domain/information models
- **Medium** granularity (architectural level, development perspective)
 - **Components** and their **interfaces**
architecture documentation
 - **Deployment**
nodes and related/deployed artifacts, connections between nodes
- **Fine** grained (low level, development perspective, implementation details)
 - Classes, Entities, Packages
 - Component internals

Models for Behavior

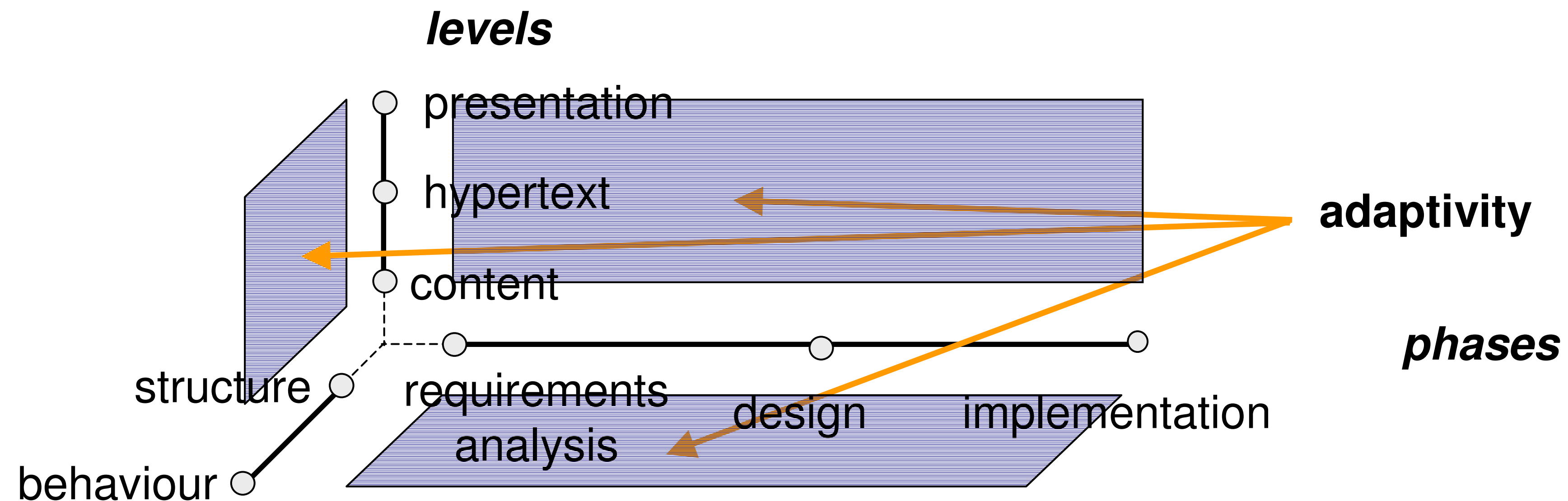
- **Coarse** grained behavior (high level, requirements)
 - **use cases** for different **roles** and actors
 - scenarios and **sequence** of actions
 - **business processes**
- **Fine** grained behavior (as a specification)
 - **state** of objects and state changes
 - **activities** in process tasks
 - **sequence** of actions
 - **timing**

Specific Models for Web Applications

- Web Applications require **specific models**
- Aspects that have to be addressed are
 - Content
 - Navigation
 - Presentation
 - Processing
- A cross-cutting aspect, not only in web applications, are **security features, roles and permissions**.
- We address this aspect later in this course!
- References for further reading
 - UMLSec - Security extensions to UML
 - RABAC - Role And Attribute-Based Access Control
 - Lecture “Advanced topics of Web and Data-intensive systems and their security”

Dimensions of Web Modeling

- Adaptivity is relevant for all dimensions (depicted as planes)

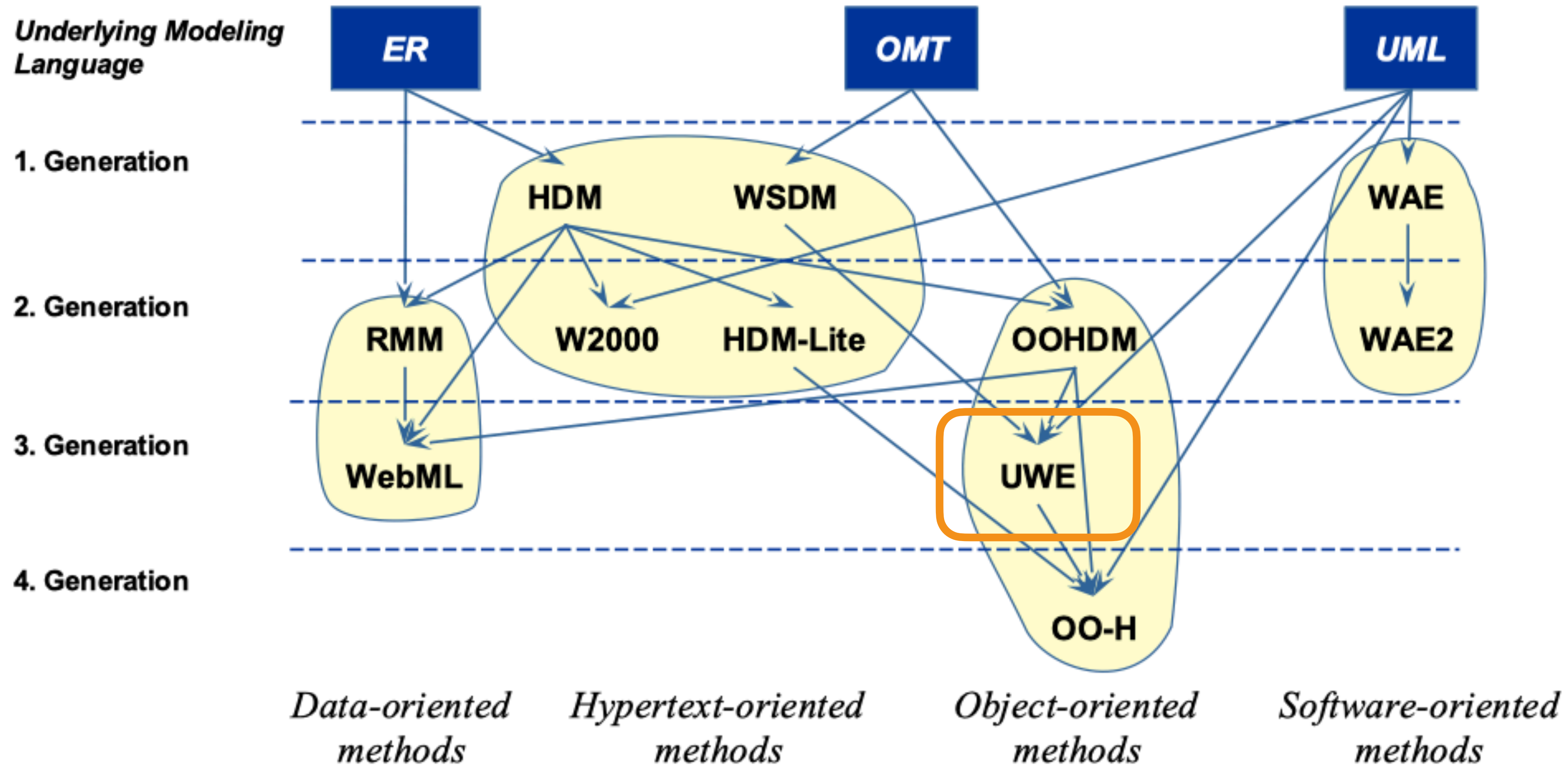


[http://uwe.pst.ifi.lmu.de/publications/MDWE-UWE_URJC_280508.pdf]

Customization and Adaption

- In recent web applications, each aspect is subject to **customization** and **adaption**.
- The application provides specific services and/or views to its users depending on the **current context**.
- This takes various **context properties** into account, such as
 - device type
 - network bandwidth
 - user identity
 - location
 - preferences
 - previous actions
 - social links
 - ...

Historical Development of Modeling Methods for the Web



UWE - UML based Web Engineering

- Overall goal of the UWE approach:
Employ methods of MDE to develop web applications
- The **UWE** approach aims at re-using as much as possible from the UML standard, while providing **domain-specific profiles and extensions** for web applications.
- Profiles are light-weight extensions to UML
- UWE comes with tooling and **automatic transformations** (as a research prototype)
- Tool support exists for design and code generation

For more details, please refer to
<https://uwe.pst.ifi.lmu.de>

UWE: Technical Aspects

- UWE supplies
 - a UML profile (version 2.1): lightweight extension using stereotypes
 - a meta model
 - a model-driven development process
 - tool support for editing UWE models (magicUWE, plugin for magicDraw) and
- (semi-)automatic generation of web applications (based on ATL and graph transformations)
[uwe.pst.ifi.lmu.de/toolMagicUWE.html]

UWE: Material / Literature

- The UWE web site contains
 - many examples: [uwe.pst.ifi.lmu.de/examples.html]
 - a tutorial: [uwe.pst.ifi.lmu.de/teachingTutorial.html]
 - the metamodel and profile [uwe.pst.ifi.lmu.de/download/UWE-Metamodel-Reference.pdf]
 - Literature [http://uwe.pst.ifi.lmu.de/publicationsByTopic.html], e.g.:
 - N. Koch, A. Kraus: The Expressive Power of UML-based Web Engineering,

Proc. of the 2nd International Workshop on Web-Oriented Software Technology (IWWOST 2002). Malaga, Spain, June 2002.

- N. Koch: Model-Driven Web Engineering UWE Approach, Universidad del Rey Juan Carlos (invited talk), Madrid, 28.05.2008
http://uwe.pst.ifi.lmu.de/publications/MDWE-UWE_URJC_280508.pdf

(the next slides are part of that document, due to copyright, please download the slide deck to learn the details)

Summary: Models

- Content
 - Domain Model
 - User Model
 - Navigation node specific composition of domain objects (cf. DTO pattern)
- Navigation
 - Nodes and navigation links
 - Several possible representations
e.g. class diagram (UWE), state machine (JSF)
 - Transformation to anchors and links in web pages, relation to business logic, e.g. for queries
- Presentation
 - UI Components, Layout
 - composition of navigation nodes
 - Transformation to specific technologies, e.g. HTML5, AngularJS, React
- Process (not specific for web applications)
 - Sequence of actions, actors
 - Relation to business logic
- Adaptivity
 - Context Properties, User Model
 - Adaption of content, navigation links, and presentation components

What we have learned...

Modeling (Part I)

- ✓ Purpose of Models
- ✓ Types of Models
- ✓ Model Driven Engineering
- ✓ Basics of the UWE approach

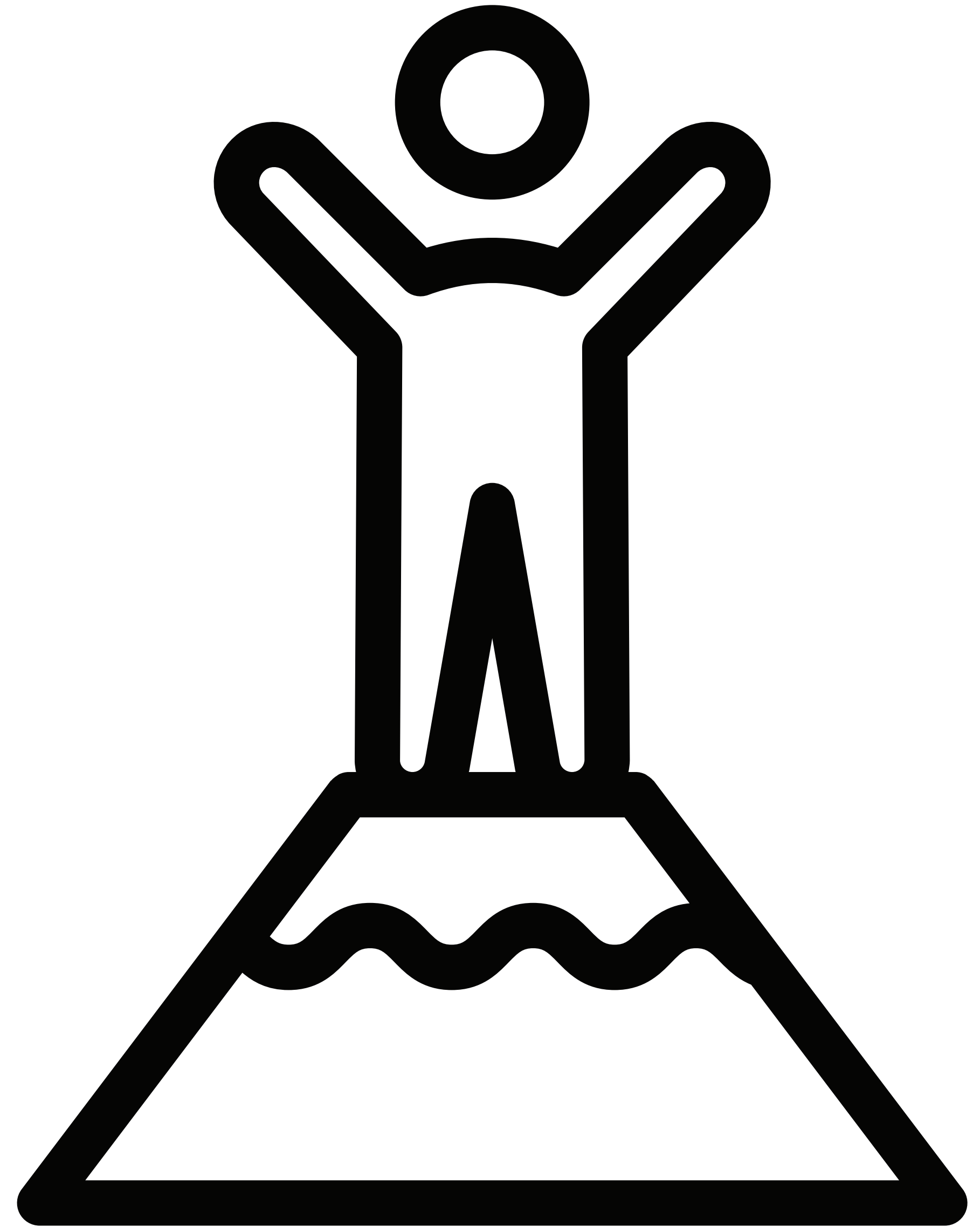


Image: colourbox.de