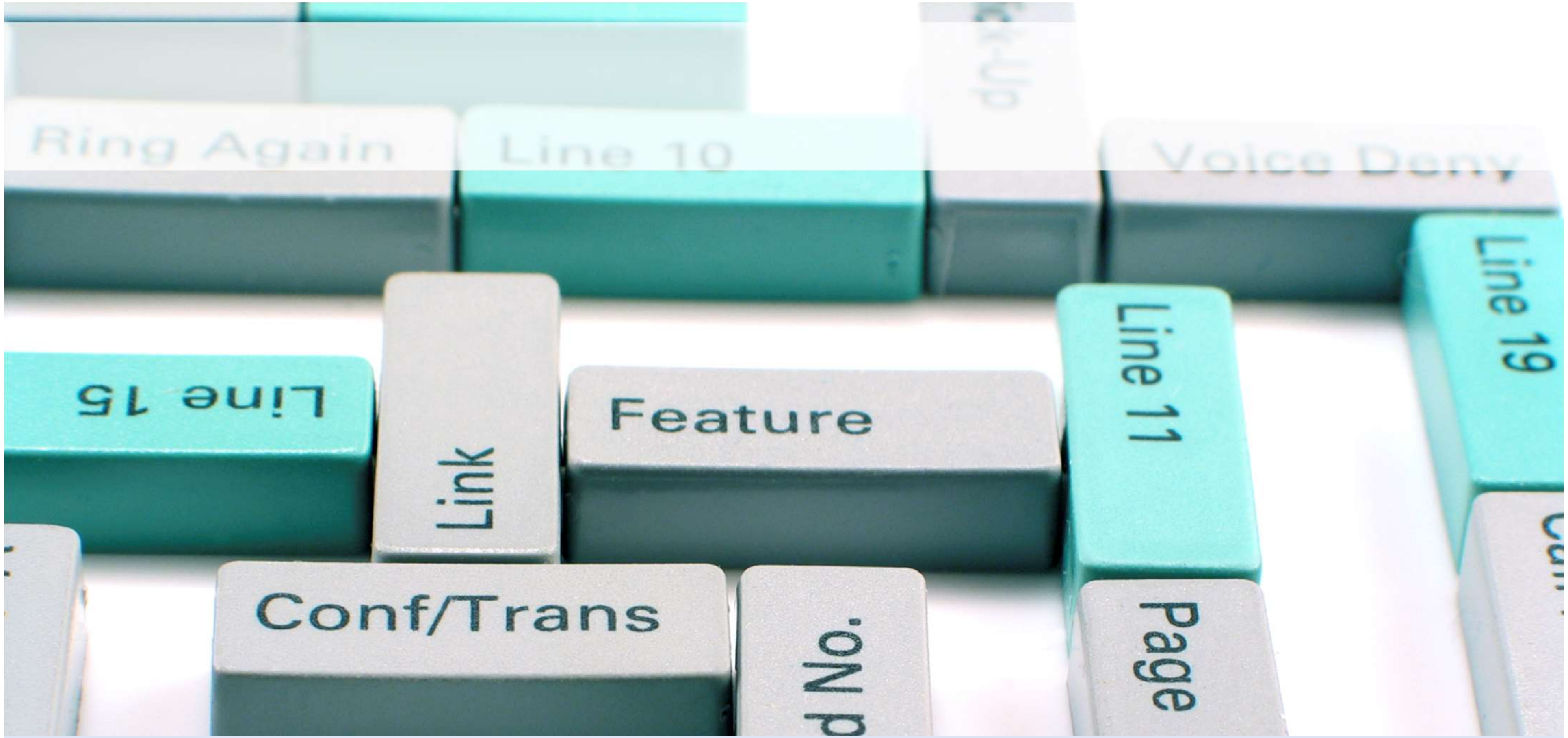


Object Oriented Analysis and Design

Prof. Dr. Robert Mertens

Introduction and Definitions

Divide and Conquer

… or: „What is the secter of Object Oriented?“

- Keep it simple
  - Objects solve partial problems
  - Those are easier to describe
- Keep it close to the problem domain
  - Objects represent algorithmic steps
  - Objects represent real entities
- Allocate responsibilities
  - objects
  - … solve problems
  - … use other objects

„Object oriented design is like teamwork: Everyone knows their task. And inividual tasks are easier to desribe than the process as a whole!“

Definition: Analysis vs. Design

… or: „The right thing (analysis) the right way (design)!“

Definition

Analysis

- Understanding problem and domain
- What are the requiremens?
- What objects are in the domain?

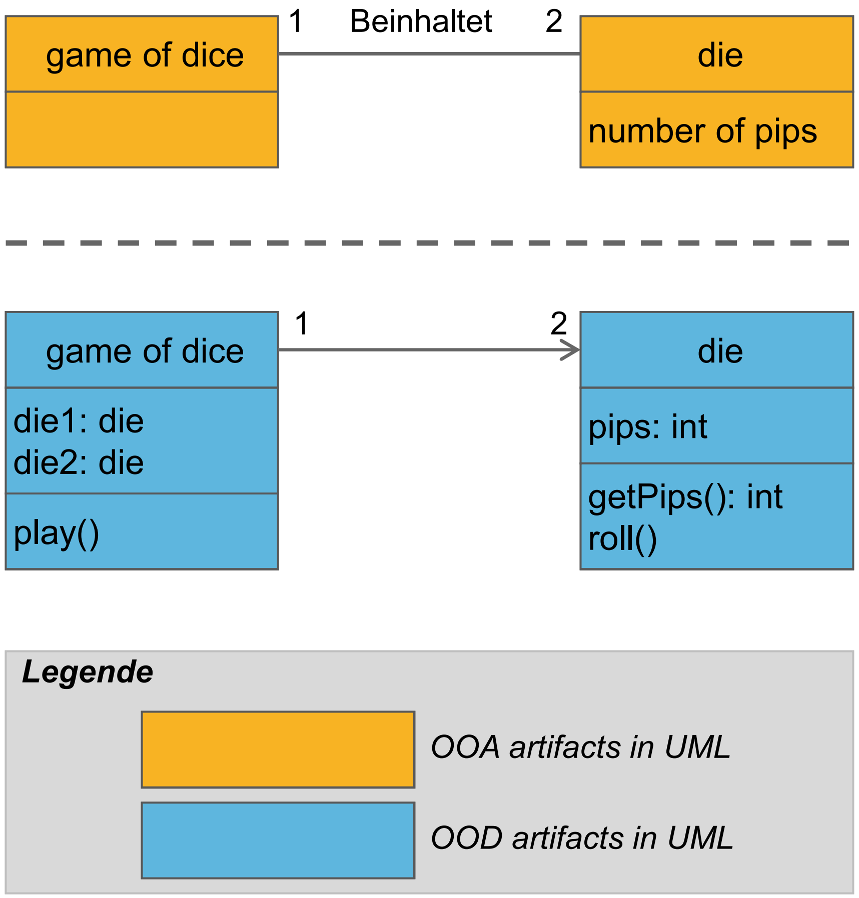
Design

- Designing a solution
- Concept, not implementation
- No obvious details

Object Oriented Analysis and Design

… oder: „What happens at which stage?“

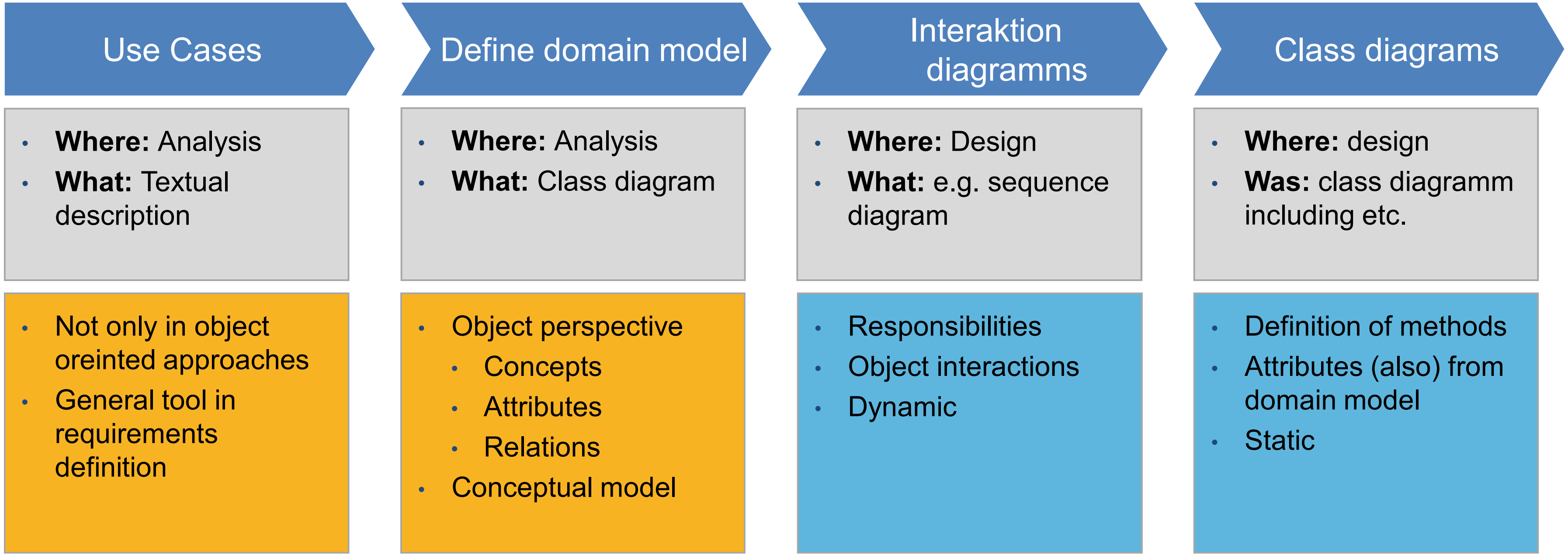
- Object oriented analysis (OOA)
  - In application domain
  - Domain objects and their relations
  - Fining concepts
- Object oriented design (OOD)
  - Definie software objects
  - Define methods
- Different types of artifacts



An example

Stages and Artifacts

… oder: „How to use OOA/OOD?“



Use Cases	Define domain model	Interaktion diagramms	Class diagrams
• **Where:** Analysis • **What:** Textual description	• **Where:** Analysis • **What:** Class diagram	• **Where:** Design • **What:** e.g. sequence diagram	• **Where:** design • **Was:** class diagramm including etc.
• Not only in object oreinted approaches • General tool in requirements definition	• Object perspective • Concepts • Attributes • Relations • Conceptual model	• Responsibilities • Object interactions • Dynamic	• Definition of methods • Attributes (also) from domain model • Static

Use Case Game of Dice

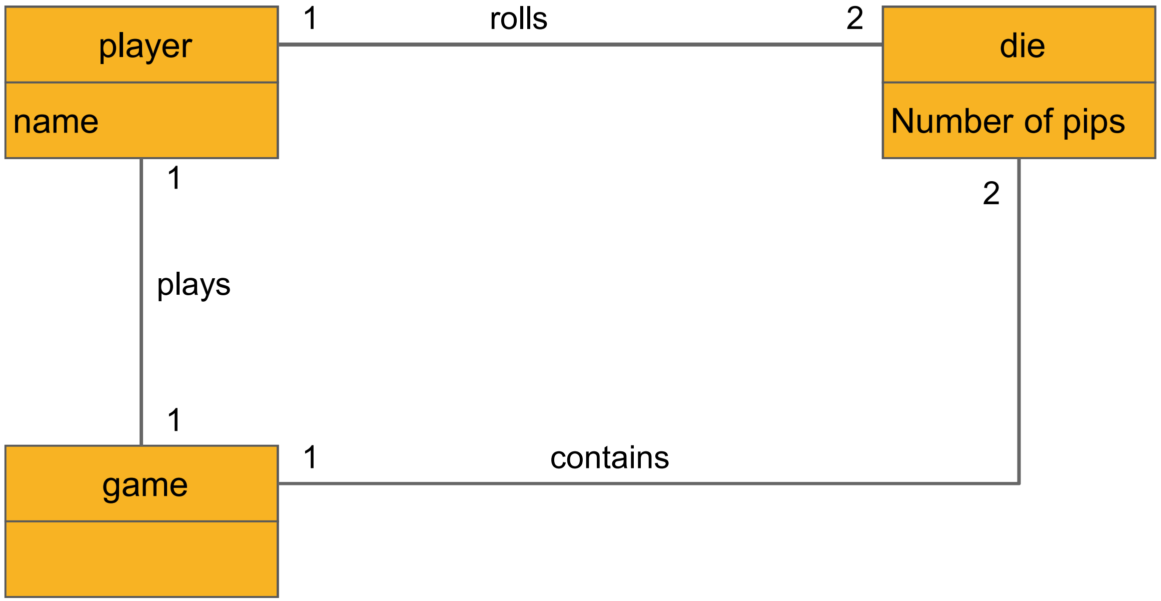
… or: „Textual description of the application secenario.“

- Text
  - Brief story
  - Only important aspects
- Best case as starting point
- Expand with alternatives
  - Errors
  - Variations (like input methodss)

*„**A game of dice:** A player starts a new round. The system shows the result: 7: player wins, else player looses. “*

Domain Model for Game of Dice

… oder: „How does the program see the world?“



„A domain model shows contains all relevant concepts and objects from the application domain. “

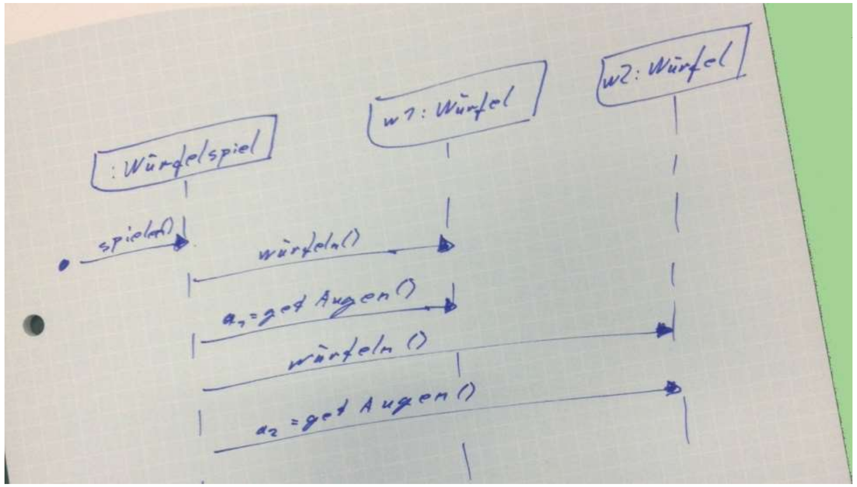
Interaction Diagramm for the Game of Dice

… oder: „What happesn when the program is running?“

- Example runs
  - Not complete program logic
  - Standard cases
  - Problem cases

- Different types of diagrams
  - Sequence diagrams
  - Communication diagrams
  - etc.

- Difference domain ↔ program
  - Player rolls dice ↔ system does
  - System is not a direct model



„Sequence diagrams are ideal for visualizing concroete sequences of actions. Avoid exceptions and conditions as they quickly lead to clotted diagrams!“

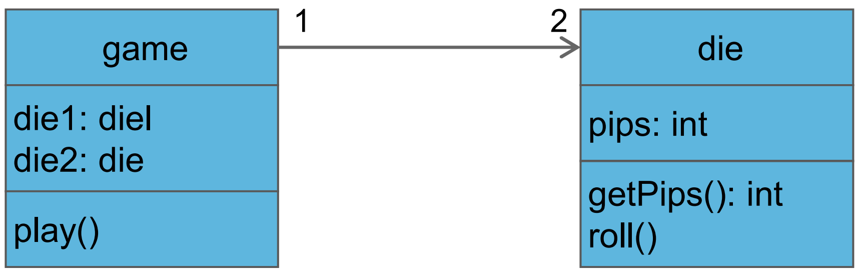
Classes in the Software

… or: „What‘s under the hood?“

- Whom do the objects know?
  - Object relations
  - Allowing to call other objects‘ methods

- What do they know?
  - Attributes
  - Can be uses in methods

- What can they do?
  - Described in methods
  - Use attributes
  - Call (other objects‘) methods



game

die1: diel
die2: die

play()

1

2

die

pips: int

getPips(): int
roll()

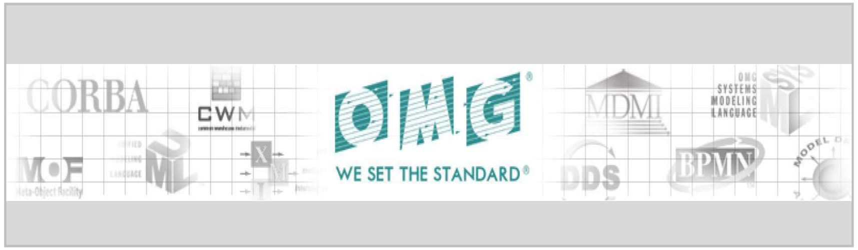
„Class diagrams convey a lot of information in an accessible way. The are hence a commonly used form of visualization.“

Die Unified Modeling Language
UML

Die Unified Modeling Language

… or: „What exaclty is UML?“

- Visual language
  - Diagram notation
  - Many kinds of diagrams
- Profiles for application scenarios
  - E.g. *EJB profile*
- Semantics
  - Described in UML Metamodell
  - Can be translated to code



„The Unified Modeling Language is a visual language for specifying, constructing and documenting the artifacts of systems.“

UML 2.0 Infrastructure Specification.
www.omg.org. 2003.

Application strategies

… or: „And what is it good for?“

Nutzungsmöglichkeiten der UML

sketch

- Visualizing only parts
- Discussing problems
- Discussing solutions

blueprint

- Reverse engineering/understanding code
- Generating code (to be completed/modiefied by developers)
- Both with support by tools

programming

- Translates into executable code
- i.e. coding program logic
- Still a resuearch topic

„No Silver Bullet: UML is a notation for diagrams not a magic wand. UML is a great communication and visualization tool, but it does not replace software design skills! “

UML modeling perspectives

… or: „What exactly do we describe with UMS?“

- One notation – three perspectives
- Similar class names on all levels
  - Closing representational gaps
  - Facilitating understanding
- Two software oriented perspectives
  - Specification and implementation
  - Usually only implementation perspective

conceptual perspektive

Describes the real world / problem domain

specification perspective

Describes software components with specifications and interfaces

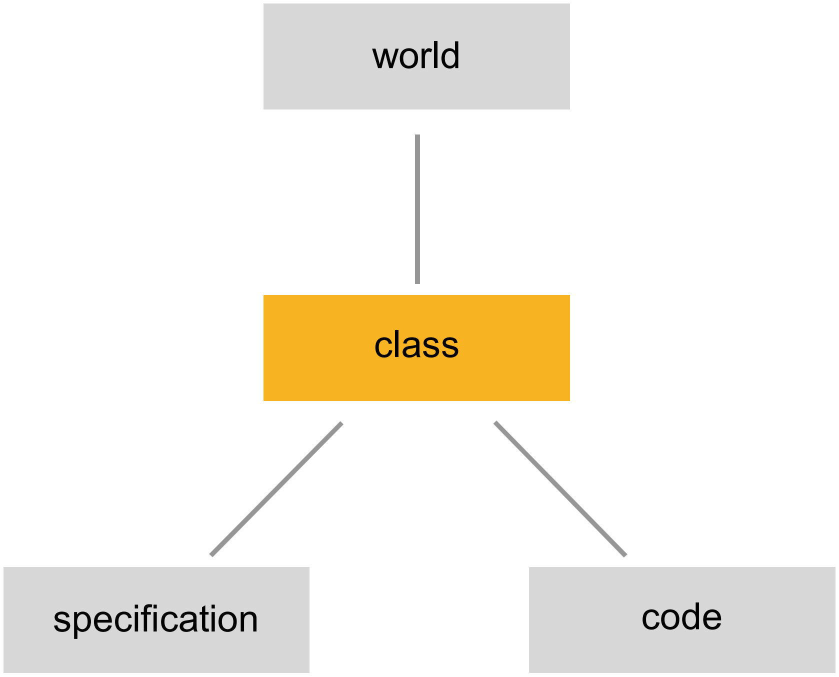
implementation perspective

Like specification perspective but with datatypes and technology specific information

The Term *Class* in Three Perspectives

… or: „What does *class* mean?“

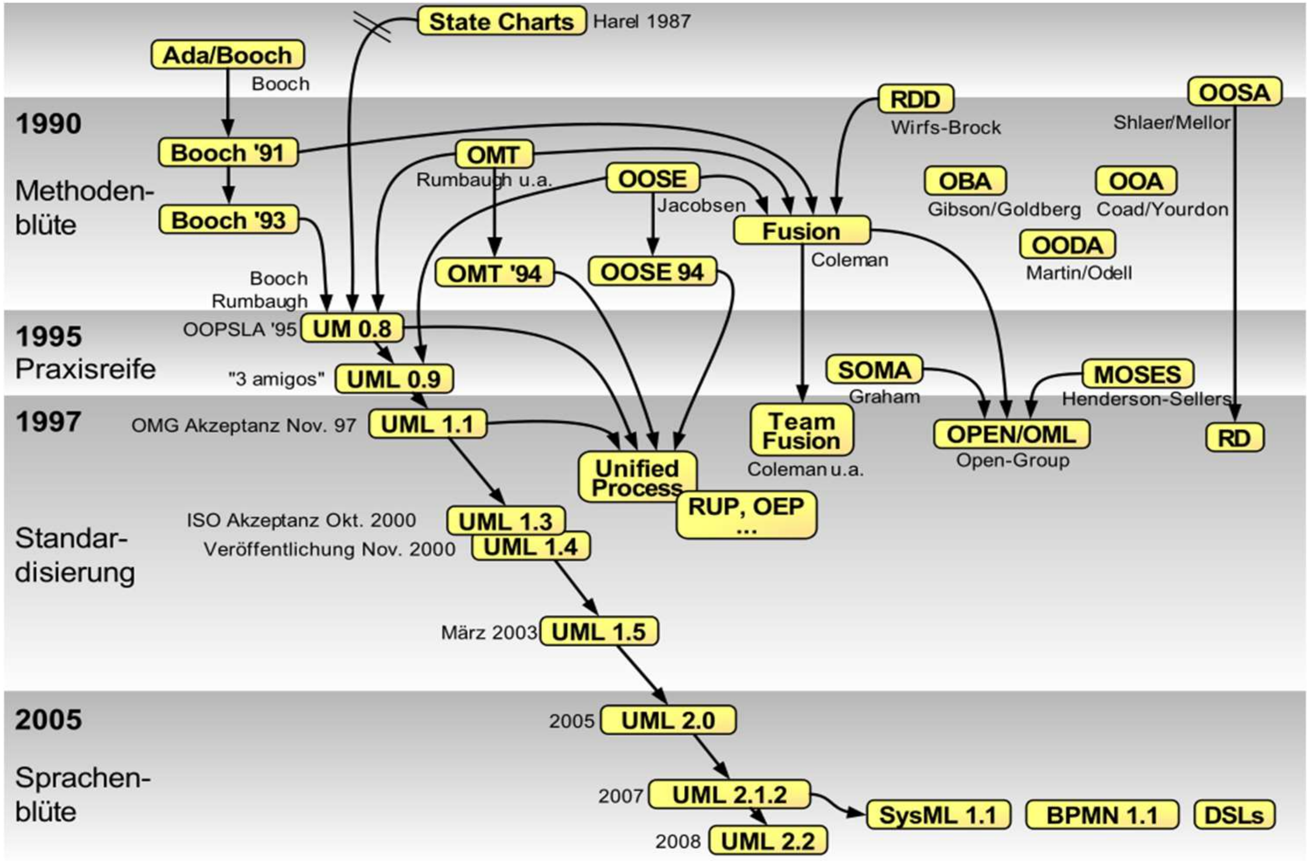
- conceptual Class
  - Concept or thing in the real world
- Software class
  - Description in specification or implementation perspective
- Implementation class
  - Concrete implementation
  - Java-Code, C#-Code, etc.



„The term class cann be used in three different contexts “

A little history on UML

… or: „Why *unified*?“



http://en.wikipedia.org/wiki/File:OO-historie-2.svg Autor: Axel Scheithauer

Thank you for your attention!