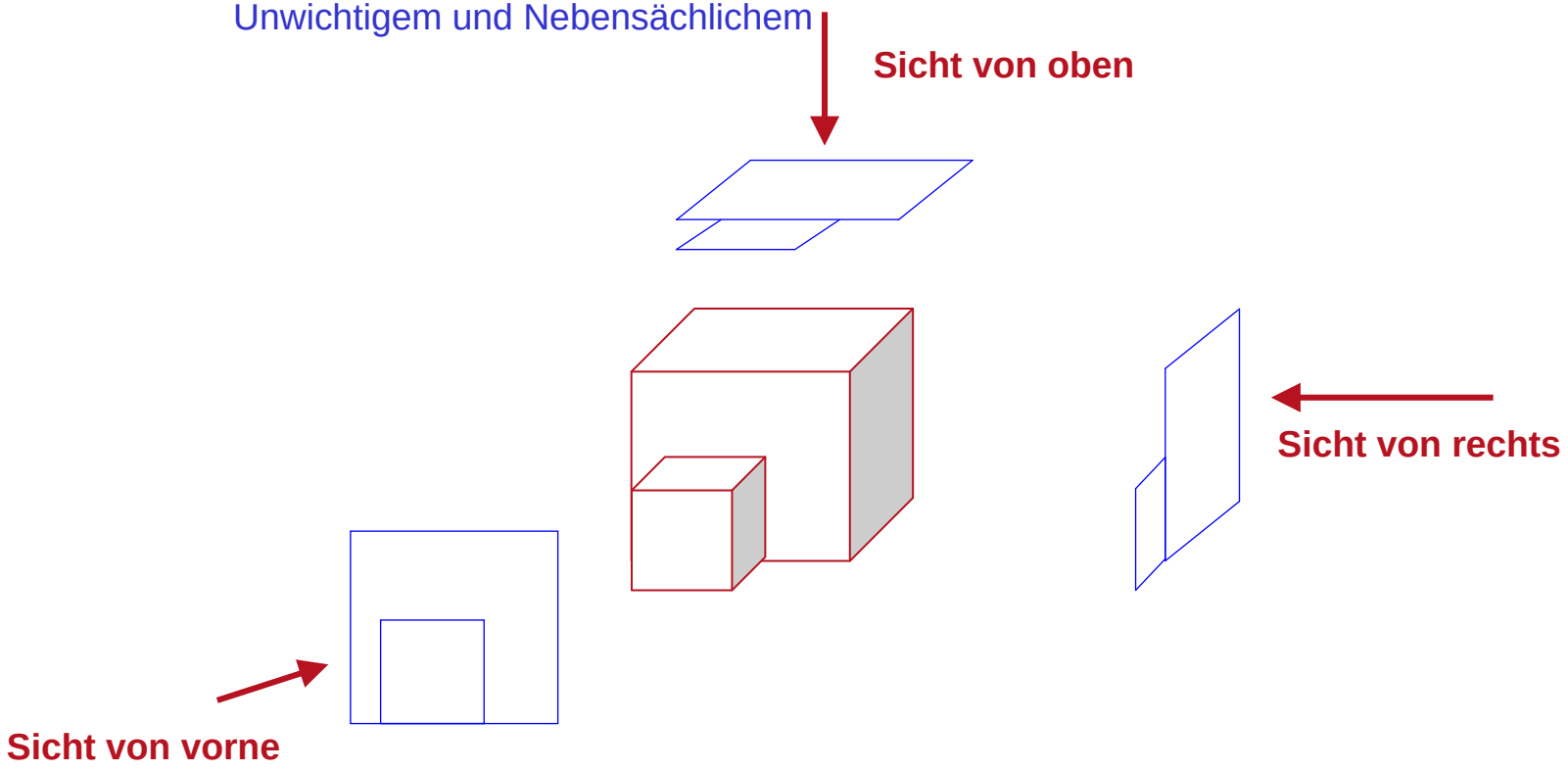
Grundlegendes

Modellierung

- Was sind die wesentlichen Gedanken?
  - Verringerung der Komplexität durch Hervorheben des Wesentlichen und Weglassen es Unwichtigen
  - Abstraktion der Realität aus spezifischen Sichtweise(n)
  - Abstraktion bedeutet Verringerung der Komplexität durch Vernachlässigung von Unwichtigem und Nebensächlichem



Grundlegendes

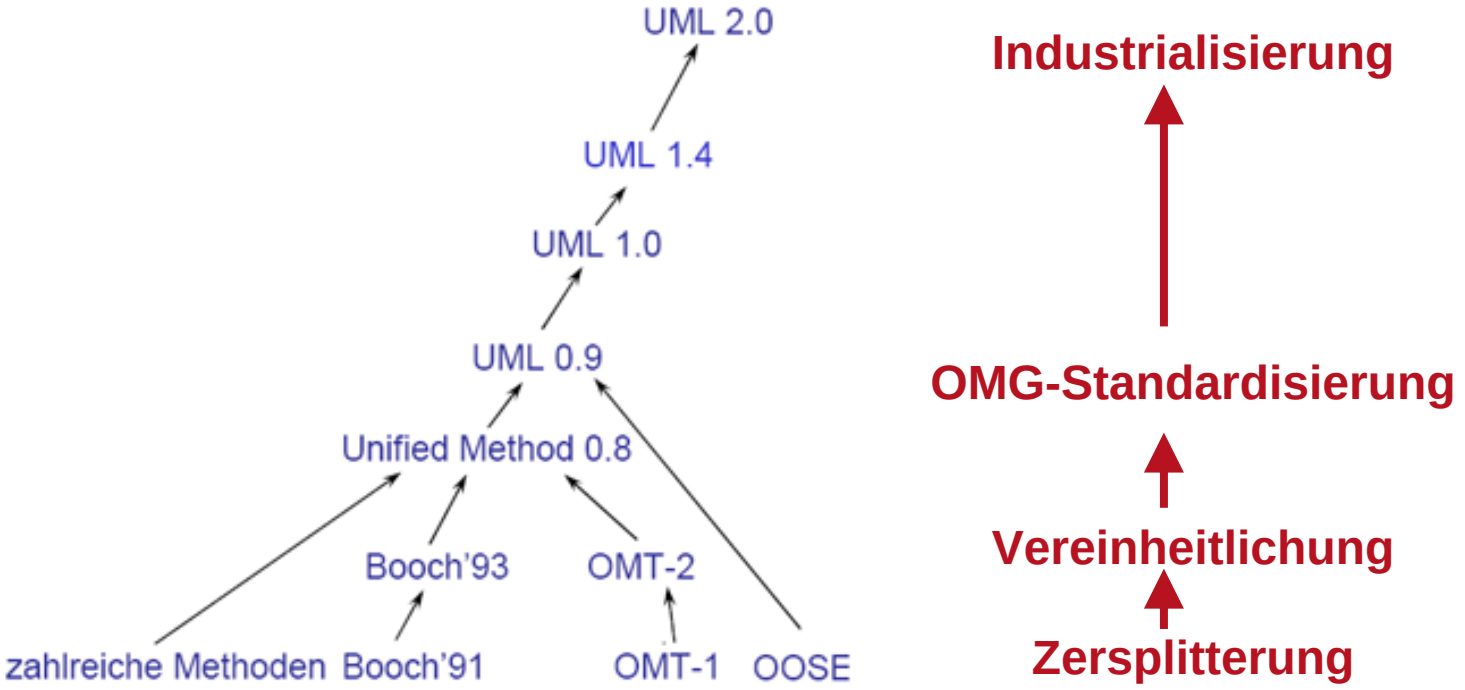
Modellierung

- Warum überhaupt?
  - Visualisierung
    - Vereinfachte Darstellung einer oft komplexen Realität
    - Ein Bild sagt mehr als 1000 Worte
  - Kommunikation
    - Basis einer Kommunikation aller Beteiligten durch definierte Standards
  - Überprüfbarkeit
    - Auf Vollständigkeit, Widerspruchsfreiheit und Korrektheit
    - insbesondere durch Darstellung von Beziehungen
- Einflussfaktoren
    - Notwendiger Detailierungsgrad (Wie detailliert/kompliziert muss/darf es sein?)
    - Vorhandenes Know-How der Prozesse
    - Verfügbare Zeit
    - Welche Sprache und auch Fachausdrücke können verwendet werden?
    - Welche Abstraktionsebene soll gewählt werden? (niedrige Abstraktionsebene bedeutet anschaulicher, höhere bedeutet besser wieder verwendbar)

Grundlegendes

Definitionen

- UML (Unified Modeling Language) ist ein Standard der OMG (http://www.omg.org/uml)
  - UML ermöglicht es auch komplexe Systeme in einfachen Worten und Bilder zu beschreiben
  - ist keine Methode, sondern definiert eine standardisierte Notation und Semantik zur Visualisierung, Konstruktion und Dokumentation von Modellen für die Geschäftsprozessmodellierung und für die objektorientierte Softwareentwicklung.



Grundlegendes

Zusammenhang mit MDA

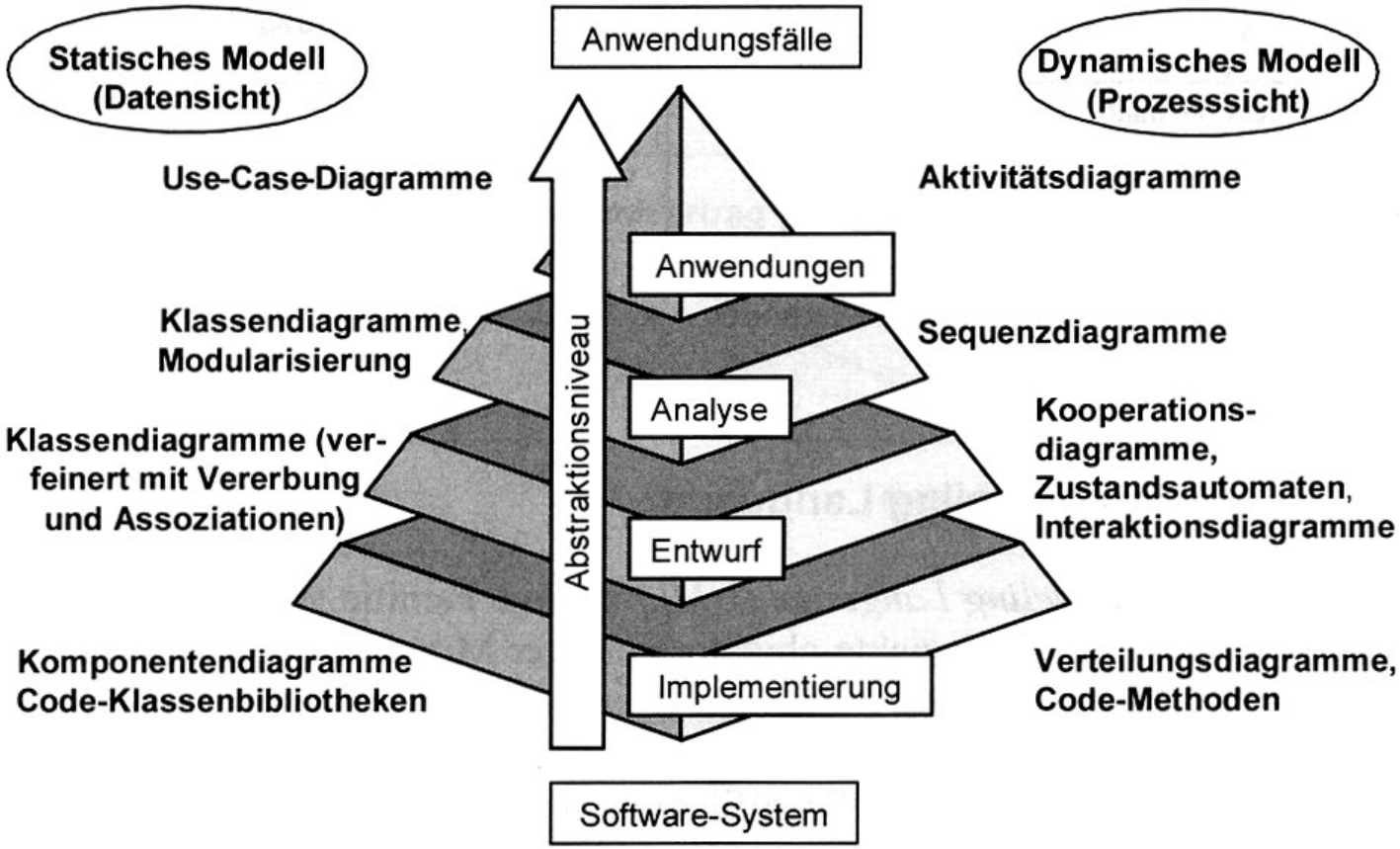
- MDA (Model Driven Architecture)
  - MDA ist ebenfalls ein Standard der OMG (http://www.omg.org/mda)
  - MDA definiert eine Vorgehensweise beim Softwareentwicklungsprozess unter Verwendung von UMLs
  - MDA unterscheidet zwischen PIM (Platform Independend Model), PSM (Platform Specific Model) und Code und bietet eine Grundlage für die automatische Code-Generierung
    - MOF, XMI - Die UML-Metadaten der OO-Modelle werden in MOF-Repositories (Meta-Object Facility) gespeichert und können per XMI (XML Metadata Interchange) mit anderen UML- oder MDA-Tools ausgetauscht werden
    - PIM, PSM - Die Modellierung der fachlichen Geschäftsprozesse (PIM, Platform Independend Model) ist von der Modellierung der technischen Realisierung (PSM, Platform Specific Model) getrennt.
    - Austauschbare technische Plattform - Die späte Wahl bzw. die spätere Umentscheidung der verwendeten technischen Plattform (z.B. J2EE, .NET) ist möglich.
    - Patterns und Wiederverwendung - Bewährte Design Patterns, Best Practices und wiederverwendbare Komponenten werden unterstützt.
    - Vorgehensmodelle - MDA ist mit etablierten Entwicklungsprozessen und Vorgehensmodellen vereinbar (z.B. RUP, Agile Modeling, XP).
    - Automatisierung - Sowohl die Transformation des PIMs in das PSM (z.B. mit J2EE Patterns) als auch die anschließende Generierung von Sourcecode (z.B. Java) kann eventuell (zumindest weitgehend) automatisiert werden (z.B. für Webanwendungen).

Wurzeln

- OMT - Object Modeling Technique (James Rumbaugh et al.)
  - Analyse datenintensiver Informationssysteme
  - insbesondere Verwendung von erweiterten Entity-Relationship-Diagrammen
- Booch-Method (Grady Booch)
  - Modellierung von Echtzeitsystemen und nebenläufigen Systemen
  - beeinflusst durch Programmiersprachen (insb. Ada)
- OOSE - Object-Oriented Software Engineering (Ivar Jacobson)
  - Use Case-orientierter Ansatz
  - insbesondere geeignet für die Modellierung von Geschäftsprozess
    - Ein Geschäftsprozess besteht aus einer Anzahl von Aktivitäten bzw. Tätigkeiten, welche in einer definierten Reihenfolge durch Personen oder Maschinen zur Erreichung eines definierten Zieles ausgeführt werden müssen

UML-Diagramme

Übersicht über die Gängigsten



Quelle: Bodendorf, F.: Daten- und Wissensmanagement, 2006

UML-Diagramme

Übersicht über die Gängigsten

- Behavior Diagrams (Verhaltensdiagramme)
  - Use Case Diagram (Use-Case-Diagramm, Anwendungsfalldiagramm)
    - stellt Beziehungen zwischen Akteuren und Anwendungsfällen dar
  - Activity Diagram (Aktivitätsdiagramm)
    - beschreibt Ablaufmöglichkeiten, die aus einzelnen Aktivitäten/Schritten bestehen
  - Statechart Diagram (Zustandsdiagramm, Zustandsautomat)
    - zeigt eine Folge von Zuständen eines Objekts
- Interaction Diagrams (Interaktionssdiagramme)
  - Sequence Diagram (Sequenzdiagramm)
    - wichtigstes Interaktionsdiagramm: zeigt den zeitlichen Ablauf von Nachrichten zwischen Objekten
  - Communication Diagram (Kommunikationsdiagramm) (früher Kollaborationsdiagramm)
    - zeigt Beziehungen und Interaktionen zwischen Objekten

UML-Diagramme

Übersicht über die Gängigsten

- Structural Diagrams (Strukturdiagramme)
  - Class Diagram (Klassendiagramm)
    - wichtigstes Diagramm: Klassen und ihre Beziehungen untereinander
  - Object Diagram (Objektdiagramm)
    - Objekte, Assoziationen und Attributwerte zu einem bestimmten Zeitpunkt während Laufzeit
  - Component Diagram (Komponentendiagramm)
    - Komponenten und ihre Beziehungen und Schnittstellen
  - Deployment Diagram (Verteilungsdiagramm)
    - Einsatzdiagramm, Knotendiagramm, Laufzeitumfeld