

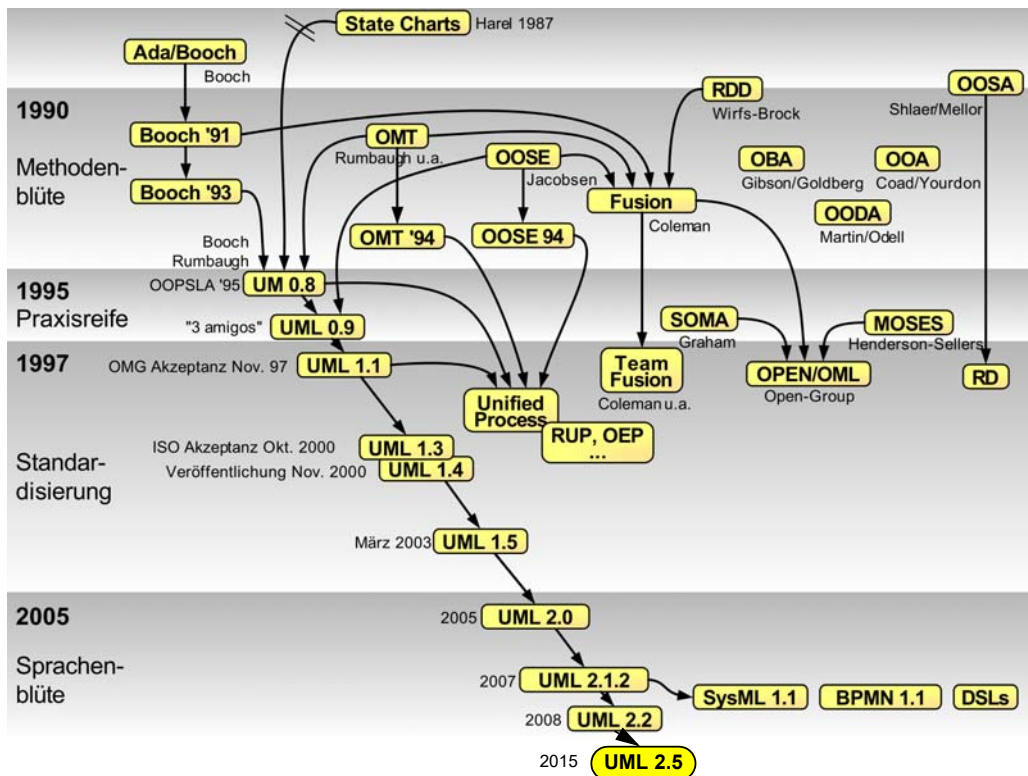
4 Objektorientierte Modellierung & UML

Inhalt

4.1	Entwicklung objektorientierter Modellierung	-2
4.2	Grundlegende Konzepte der objektorientierten Modellierung	-4
4.3	Umsetzung objektorientierter Modellierung in UML	-5
4.3.1	Wichtige UML Schemata für statische Aspekte	-7
4.3.2	Wichtige UML Schemata für Dynamische Aspekte	-8
4.3.3	Wichtige UML Schemata für softwareorganisatorische Aspekte	-9
4.3.4	Zusammenspiel wichtiger Schemata in UML	-10
4.3.5	Modelle, Diagramme und Werkzeuge	-11
4.3.6	UML 2.x Diagrammtypen	-12

4.1 Entwicklung objektorientierter Modellierung

- ❑ Objektorientierte Konzepte wurden erstmals in den 60iger Jahren in der Programmiersprache SIMULA eingesetzt.
- ❑ Verbreitung objektorientierter Programmiersprachen erst durch Smalltalk, Ada, C++ und Java (siehe Graphik in Abschnitt "2.3.4 Objektorientierte Modelle").
- ❑ Objektorientierte Analyse und Entwurfsmethoden ab Mitte 80iger.
- ❑ Einem "Methodenkrieg" folgt 1996 Aufruf der Object Management (OMG) zur Schaffung eines Modellierungsstandards (siehe Graphik in Abschnitt "2.3.4 Objektorientierte Modelle").
- ❑ 1998 wird die Unified Modeling Language (UML) 1.1 als Modellierungsstandard von der OMG akzeptiert.
- ❑ Seither Erweiterung von UML 1.1 und Etablierung von UML 2.x als zentrale, weit akzeptierte Modellierungssprache (siehe nachfolgende Graphik, [CC BY-SA 3.0 Credit-1]).



4.2 Grundlegende Konzepte der objektorientierten Modellierung

Die Welt wird durch die “Objekt-Brille” gesehen

- ❑ Objekte sind eindeutig identifizierbare Einheiten, z.B. “der Student Karl”, “der Ablauf der LVA Modellierung”
- ❑ Objekte stehen zueinander in Beziehung, z.B. Karl “nimmt teil an” LVA Modellierung; Objekte können in Teilobjekte untergliedert sein, z.B. “die 1. Vorlesungseinheit von LVA Modellierung”
- ❑ Objekte befinden sich im Laufe ihres Lebenszyklus in unterschiedlichen Zuständen, z.B. Karl “übt” und hält derzeit beim “Punktestand 25”, und weisen Verhalten auf, z.B. Karl “löst” die 1. Übungsaufgabe, dann die 2., ...
- ❑ Objekte können interagieren, z.B., Karl stellt die Frage “Wann ist Prüfung?” an den Professor; an einem Verhalten können damit mehrere Objekte teilnehmen

4.3 Umsetzung objektorientierter Modellierung in UML

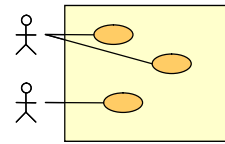
- ❑ Mit UML definiert man **Arten** von Objekten, Interaktionsszenarien, Verhaltensweisen, etc. (**Schematisierung**)
- ❑ Einzelne Objekte und Interaktionen sind darstellbar (Instanzen), aber nur wenn es ein Schema dazu gibt
- ❑ Mögliche Aussagen sind z.B.: “Alle Objekte der Art ‘Student’ stehen immer mit einem Objekt der Art ‘Universität’ in Beziehung” (‘Student’ und ‘Universität’ sind hier Schemata); “bei allen Abläufen des Verhaltens ‘XY’ passiert genau folgendes: erst ‘x’, dann ‘y’, ...”
- ❑ Nicht möglich sind Aussagen der Art: “In der Mehrheit der Fälle ...”, “So ähnlich wie...”
- ❑ Ausnahmefälle entweder ausdrücklich darstellen, z.B. in einer testbaren Spezifikation, oder ignorieren, z.B. beim Entwurf (Abschätzen, ob Sachverhalt Vernachlässigbar, ob Abstraktionsniveau für die Zielsetzung des Modells passt).

- ❑ Ein UML Modell wird i.d.R. unterschiedliche Sichtweisen auf den Problemlösungsbereich (**Aspekte**) enthalten
- ❑ Aspekte erlauben eine Fokussierung, Zusammenhänge werden klarer
- ❑ **Statische Aspekte** beschreiben Strukturen, welche über den gesamten Lebenszyklus gelten
- ❑ **Dynamische Aspekte** beschreiben die Veränderung über die Zeit (diskrete Zeit)
- ❑ **Softwareorganisatorische Aspekte** beschreiben die Untergliederung der Modelle selbst (in dieser Vorlesung nicht im Detail behandelt)

4.3.1 Wichtige UML Schemata für statische Aspekte

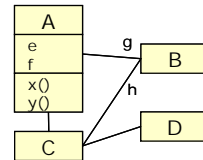
□ Anwendungsfälle

- die Benutzer und Nützlichkeit eines Systems
- definiert durch Akteure und Anwendungsfälle



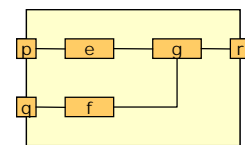
□ Klasse

- Menge gleichartiger Objekte
- definiert durch Attribute, Operationen, und Beziehungen zu anderen Klassen



□ Kompositionsstruktur

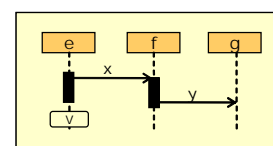
- die innere Zusammensetzung eines Objekts
- definiert durch seine Teile und die Kommunikationswege dieser untereinander bzw. mit Objekten außerhalb



4.3.2 Wichtige UML Schemata für Dynamische Aspekte

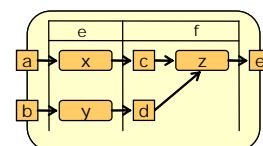
□ Interaktionen

- ge- bzw. verbotene Kommunikationsszenarien
- definiert durch Folgen von Nachrichten zwischen Interaktionspartnern



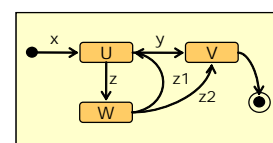
□ Aktivität

- ein in sich geschlossenes, pro-aktives Verhalten
- definiert durch Aktionen, Kontroll- und Datenfluss sowie evtl. Zuständigkeiten



□ Zustandsautomat

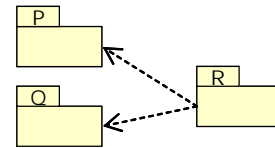
- ein reaktives Verhalten
- definiert durch Zustände, Ereignisse und Transitionen



4.3.3 Wichtige UML Schemata für softwareorganisatorische Aspekte

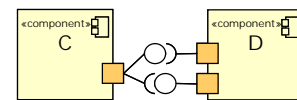
□ Pakete

- die Strukturierung einer Software in Verwaltungseinheiten und deren Abhängigkeiten
- natürlich auch für Modelle selbst



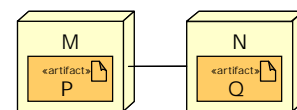
□ Komponenten

- zusammengesetzte Objekte, deren innerer Aufbau (Implementierung) stark gekapselt ist
- vereinigt Klassensicht und Paketsicht



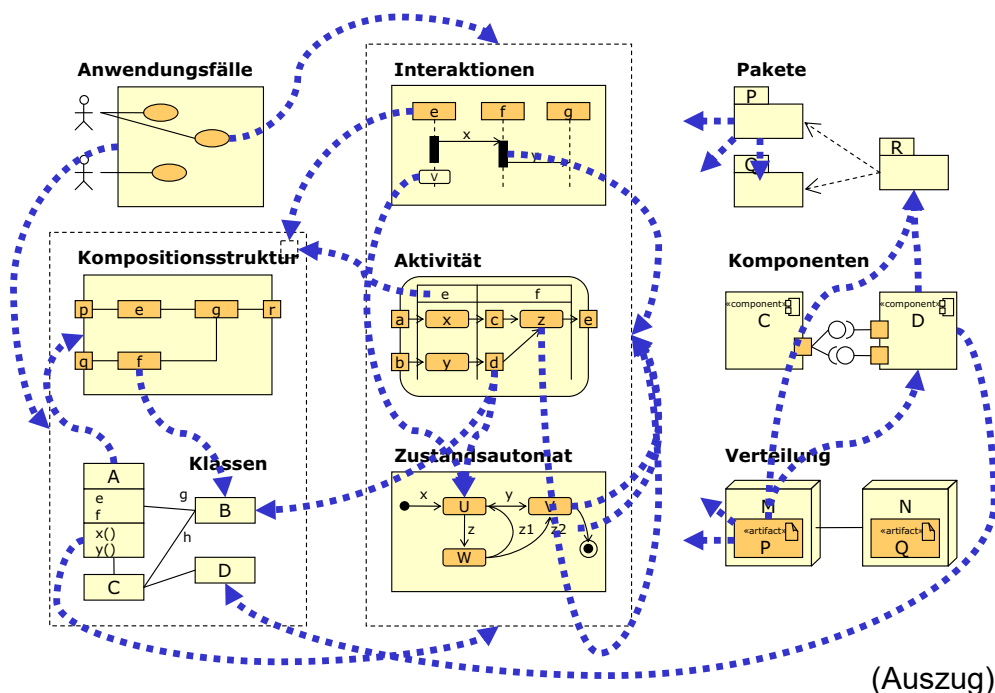
□ Verteilung

- Ausführungseinheiten und die Verteilung bzw. Installation von Software-Artefakten auf diesen



(Softwareorganisatorische Aspekte werden in dieser Lehrveranstaltung nicht im Detail behandelt.)

4.3.4 Zusammenspiel wichtiger Schemata in UML

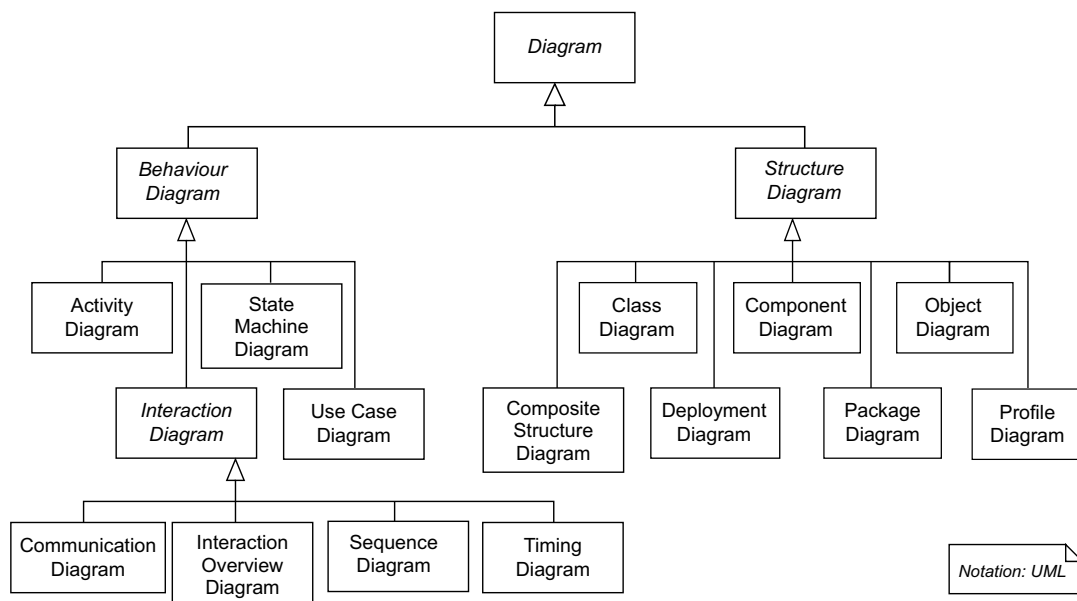


4.3.5 Modelle, Diagramme und Werkzeuge

- **Modell vs. Diagramm**
 - Ein UML-Modell ist eine Menge von Modellelementen (Klassen, Interaktionen, Zustandsmaschinen, etc.)
 - Ein UML-Diagramm ist die visuelle Darstellung von Modellelementen (eine Sicht auf ein Modell)
- **Werkzeuge**
 - Modellierungswerkzeuge implementieren die Sprache UML
Vorteil: Vorgabe der Modellstruktur, SW-Generierungsfunktionen
Nachteil: oft komplex
 - Zeichenwerkzeuge ermöglichen es nur, Diagramme zu zeichnen
Vorteil: Freiheit in der Anwendung, evtl. besser tauglich für Präsentationen
Nachteil: sehr umständlich und unzulänglich bei größeren Modellen
- **Datenaustausch**
 - Standard-Speicherformat für UML-Modelle ist XMI-basiert; Versionsproblematik
 - Austausch von Diagrammen (Bildern) i.A. unproblematisch, Bilder sind jedoch oft nicht zum Weiterarbeiten geeignet

4.3.6 UML 2.x Diagrammtypen

- **Sieben Strukturdiagramme**
 - Statische Strukturen (Klassenstrukturen bis Gliederung von ganzen Systemen und Architekturen)
 - **Klassendiagramm** ➤ **Kompositionsstrukturdiagramm**
 - **Objektdiagramm** ➤ **Verteilungsdiagramm**
 - **Komponentendiagramm** ➤ **Paketdiagramm**
 - **Profildiagramm**
- **Sieben Verhaltensdiagramme**
 - Abläufe zwischen statischen Teilen
 - **Aktivitätsdiagramm** ➤ **Interaktionsübersichtsdiagramm**
 - **Anwendungsfalldiagramm** ➤ **Kommunikationsdiagramm**
 - **Zustandsdiagramm** ➤ **Zeitdiagramm**
 - **Sequenzdiagramm**



Hierarchie zwischen den 14 UML-Diagrammen [Credit-2]



Literatur und Quellen:

- ❑ <http://www.uml.org>
- ❑ OMG Unified Modeling Language™ (OMG UML), Version 2.5
Normative Reference: <http://www.omg.org/spec/UML/2.5>
OMG Document Number formal/2015-03-01
- ❑ Martin Hitz, Gerti Kappel, Elisabeth Kapsammer, Werner Retschitzegger:
UML @ Work. Objektorientierte Modellierung mit UML 2. 3. aktualisierte und überarbeitete Auflage. dpunkt Verlag 2005. 436 Seiten. ISBN PDF: 978-3-89864-870-7
 - Die Sprache UML im Detail behandelt
- ❑ Martina Seidl, Marion Scholz, Christian Huemer, Gerti Kappel: UML @ Classroom,
DOI 10.1007/978-3-319-12742-2, eBook (ePub) ISBN: 978-3-86491-175-0 (dpunkt Verlag 2012) oder ISBN 978-3-319-12742-2 (Springer International Publishing, 2015)
 - Ein Ausschnitt von UML, speziell aufbereitet für die Lehre

- ❑ M. Blaha, J. Rumbaugh: Object-oriented Modeling and Design with UML
 - Die Methodik der Modellierung, von der Analyse bis zur Implementierung, unter Einsatz von UML
- ❑ [CC BY-SA 3.0 Credit-1] History of Object Oriented Modeling languages, Guido Zockoll, Axel Scheithauer & Marcel Douwe Dekker (Mdd) - Translation and update of File:OO-historie-2.svg by AxelScheithauer, Okt 6, 2009
- ❑ [Credit-2] https://commons.wikimedia.org/wiki/File%3AUML_diagrams_over-view.svg
By Derfel73; Pmerson (derived from: Uml diagram2.png) [Public domain],
via Wikimedia Commons from Wikimedia Commons