

Martin Glinz

Requirements Engineering I

Kapitel 6

Modellierungssprachen



Universität Zürich
Institut für Informatik

6.1 Grundlagen

- Anforderungen nicht als eine Sammlung von Sätzen in natürlicher Sprache beschreiben,
- sondern ein **anwendungsorientiertes Modell** der **Aufgabenstellung** erstellen

[Im Gegensatz dazu werden im **Entwurf** konzeptionelle und/oder physische Modelle der geplanten **Lösung** erstellt]

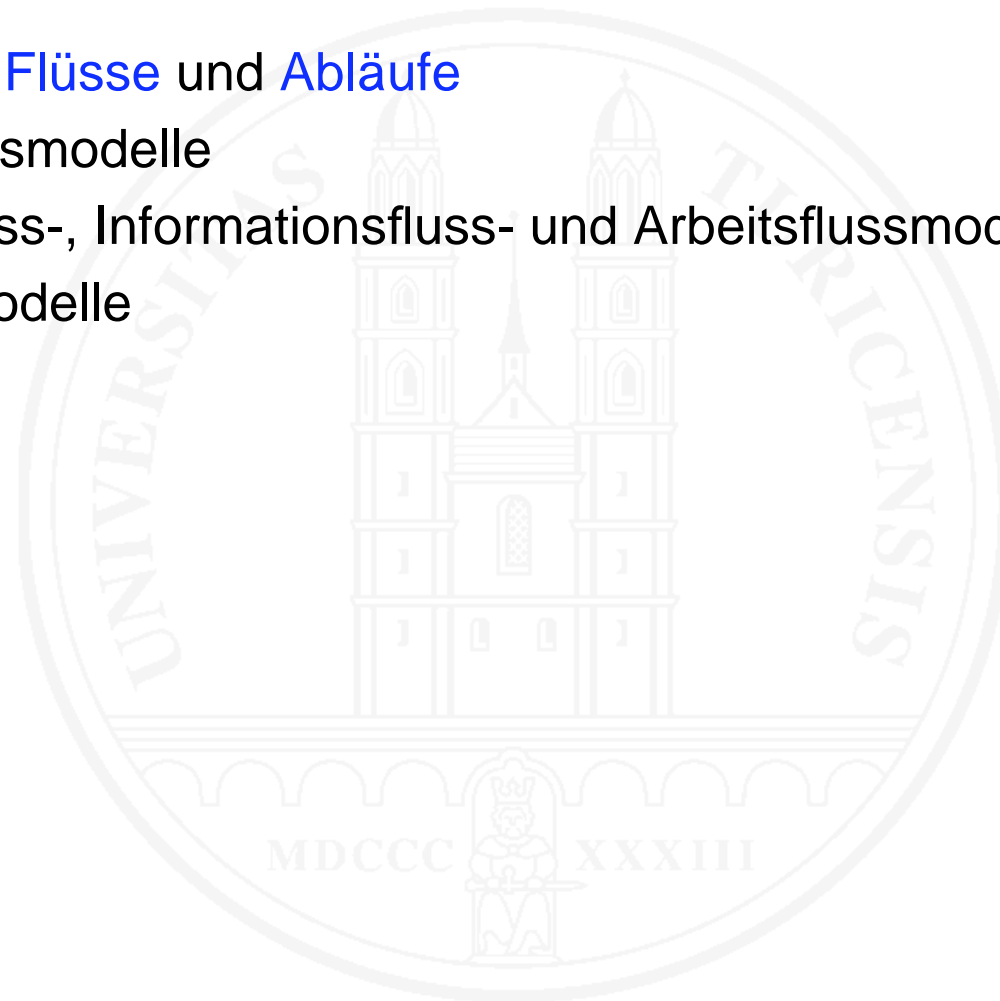
- Formale oder teilformale Modelle möglich
- Meistens (zumindest teilweise) grafisch repräsentierte Modelle
- Meistens nur für funktionale Anforderungen

Modellierungsaspekte

- **Statische Struktur**
 - Datenmodelle (Entity-Relationshipmodelle)
 - Klassen- und Objektmodelle
 - ggf. Komponentenmodelle
- **Interaktion**
 - Anwendungsfall- / Szenarienmodelle
 - ggf. Sequenz- und Zusammenarbeitsmodelle
- **Verhalten**
 - Zustandsmodelle
 - Aktivitätsmodelle

Modellierungsaspekte – 2

- Funktionen, Flüsse und Abläufe
 - Funktionsmodelle
 - Datenfluss-, Informationsfluss- und Arbeitsflussmodelle
 - Ablaufmodelle

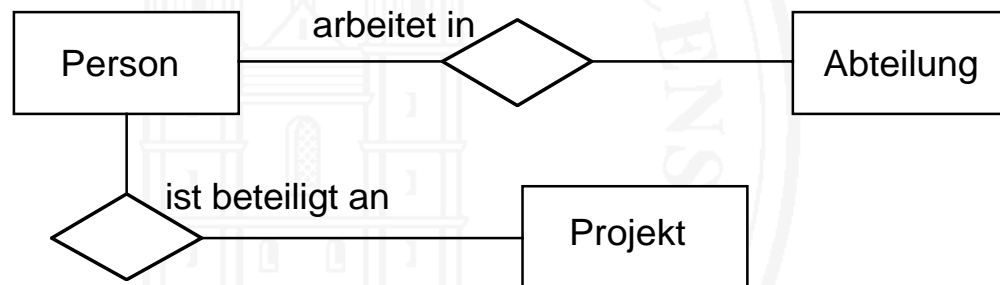


6.2 Ausgewählte Beispiele



Datenmodellierung

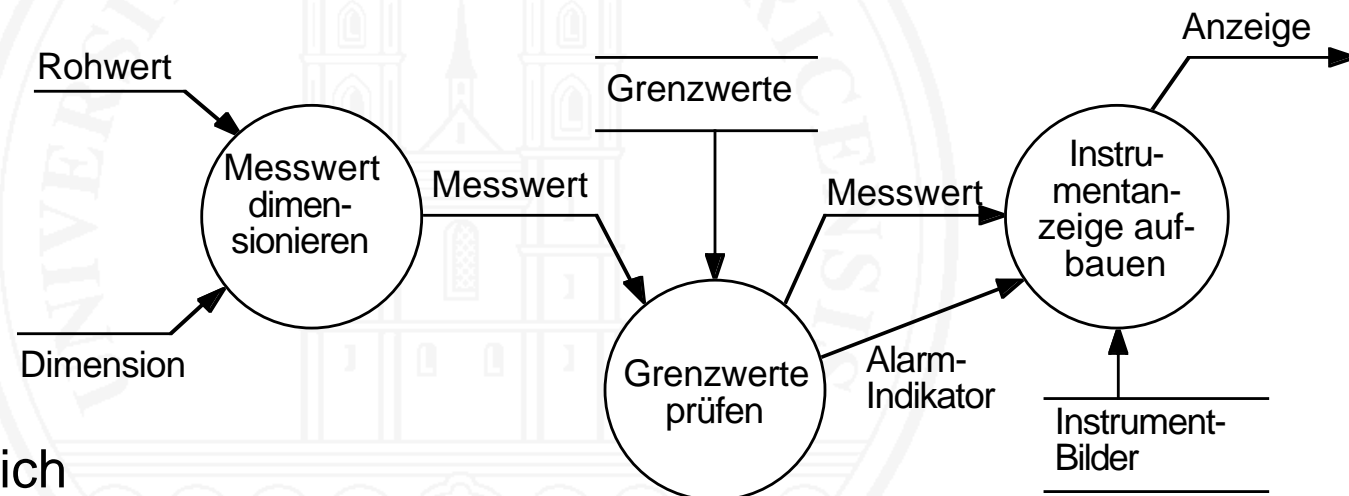
- Grundlage: **Entity-Relationship Ansatz**
- Modelliert einen Ausschnitt der Realität mit Hilfe von *Gegenstandstypen* (entity types), *Beziehungstypen* (relationship types) und *Attributen* (attributes)



- + Einfach und klar
- + Leicht auf Datenbank-Realisierungen abbildbar
- Ignoriert Funktionalität und Verhalten der Systeme
- Keine Mittel zur Systemdekomposition
- Keine Lokalität oder Einkapselung von Daten

Strukturierte Analyse

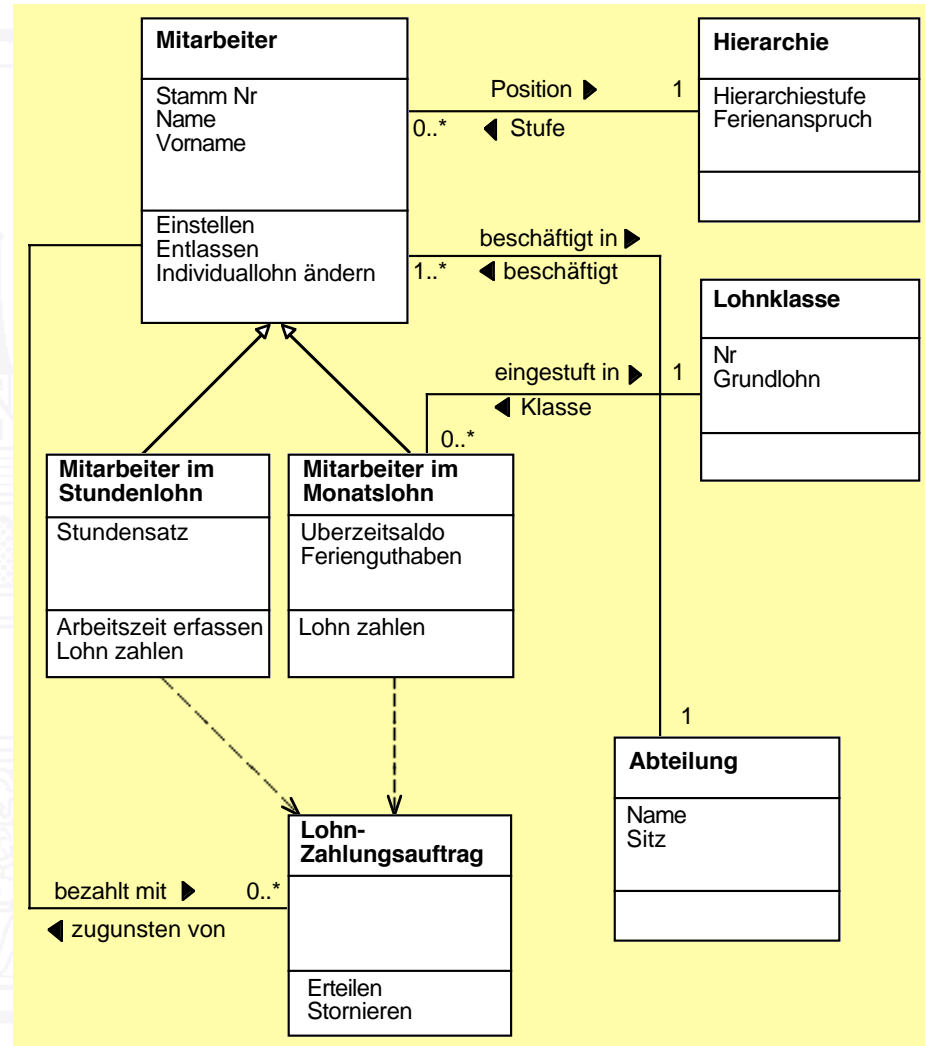
- Modelliert die Funktionalität eines Systems mit Hilfe von **Datenflussdiagrammen**
- Wird heute **eher selten verwendet**



- + Sehr anschaulich
- + Unterstützt Systemdekomposition
- Strukturbruch zwischen Spezifikation und Implementierung
- Keine Lokalität von Daten, Einkapselung nur begrenzt möglich

Objektorientierte Spezifikation

- Modelliert die **statische Struktur** eines Systems mit Hilfe von **Objekt-** oder **Klassendiagrammen**
- Näheres in Kapitel 7



Szenarien und Anwendungsfälle

- Modellieren die **Interaktion** zwischen systemexternen Akteuren und dem System
- Jede Interaktionssequenz wird durch ein **Szenario** (einen **Anwendungsfall**) beschrieben

Buch ausleihen

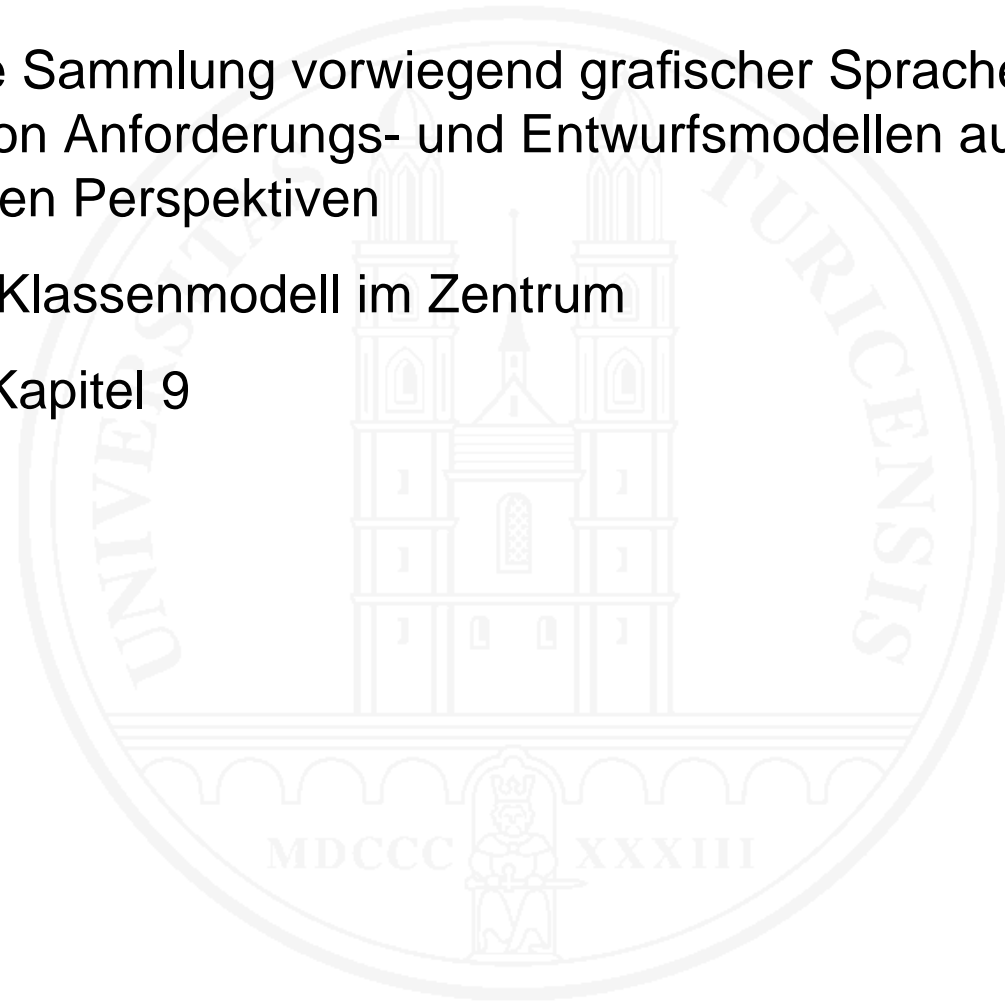
1. Ausweiskarte der Benutzerin lesen und Angaben überprüfen
2. Signatur eines Buchs lesen und zugehörigen Katalogeintrag ermitteln
3. Ausleihe registrieren und Diebstahlsicherungsetikett deaktivieren
4. ...

- Näheres in Kapitel 8

UML (Unified Modeling Language)



- UML ist eine Sammlung vorwiegend grafischer Sprachen zur Erstellung von Anforderungs- und Entwurfsmodellen aus verschiedenen Perspektiven
- Typisch ein Klassenmodell im Zentrum
- Näheres in Kapitel 9



Literatur

Cockburn, A. (2001). *Writing Effective Use Cases*. Addison-Wesley.

DeMarco, T. (1978). *Structured Analysis and System Specification*. New York: Yourdon Press.

Firesmith, D.C. (1994). Modeling the Dynamic Behavior of Systems, Mechanisms and Classes with Scenarios. *Report on Object Analysis and Design (ROAD) 1, 2* (Jul/Aug 1994). 32-36, 47.

Glinz, M. (2005). Modellierung. Skript zur Vorlesung Informatik II, Teil a. Universität Zürich.

Jacobson, I., M. Christerson, P. Jonsson, and G. Övergaard (1992). *Object-Oriented Software Engineering: A Use Case Driven Approach*. Amsterdam: Addison-Wesley.

Oestereich, B. (1998). *Objektorientierte Softwareentwicklung*. R. München: Oldenbourg.

Rumbaugh, J., Jacobson, I., Booch, G. (1999). *The Unified Modeling Language Reference Manual*. Reading, Mass., etc.: Addison-Wesley.