University of Basel

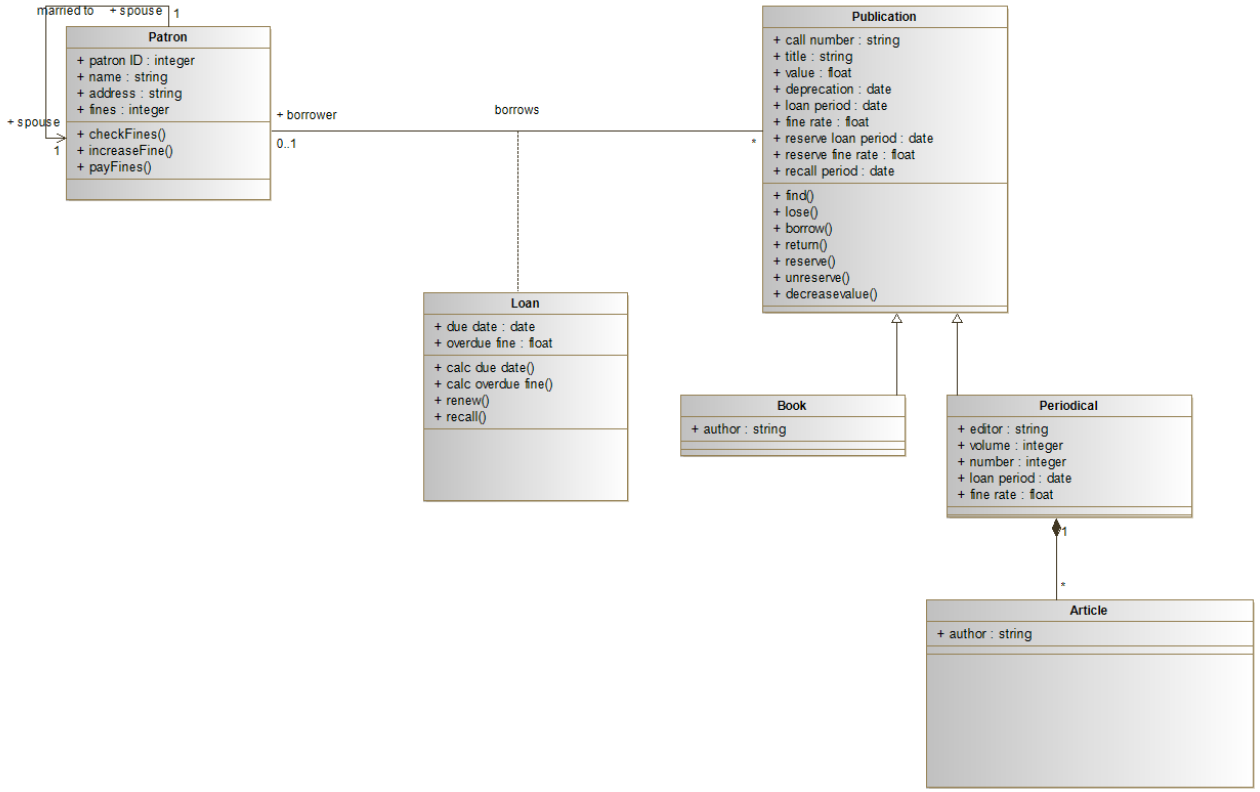
Software Engineering

Marcel Lüthi, Universität Basel

Spezifizieren mit der UML

Erinnerung: UML Klassendiagramme

Erlauben Software auf hoher Abstraktionsebene zu verstehen



Hilft beim

- Modellieren
- Dokumentieren
- Kommunizieren

Diagrammarten der UML

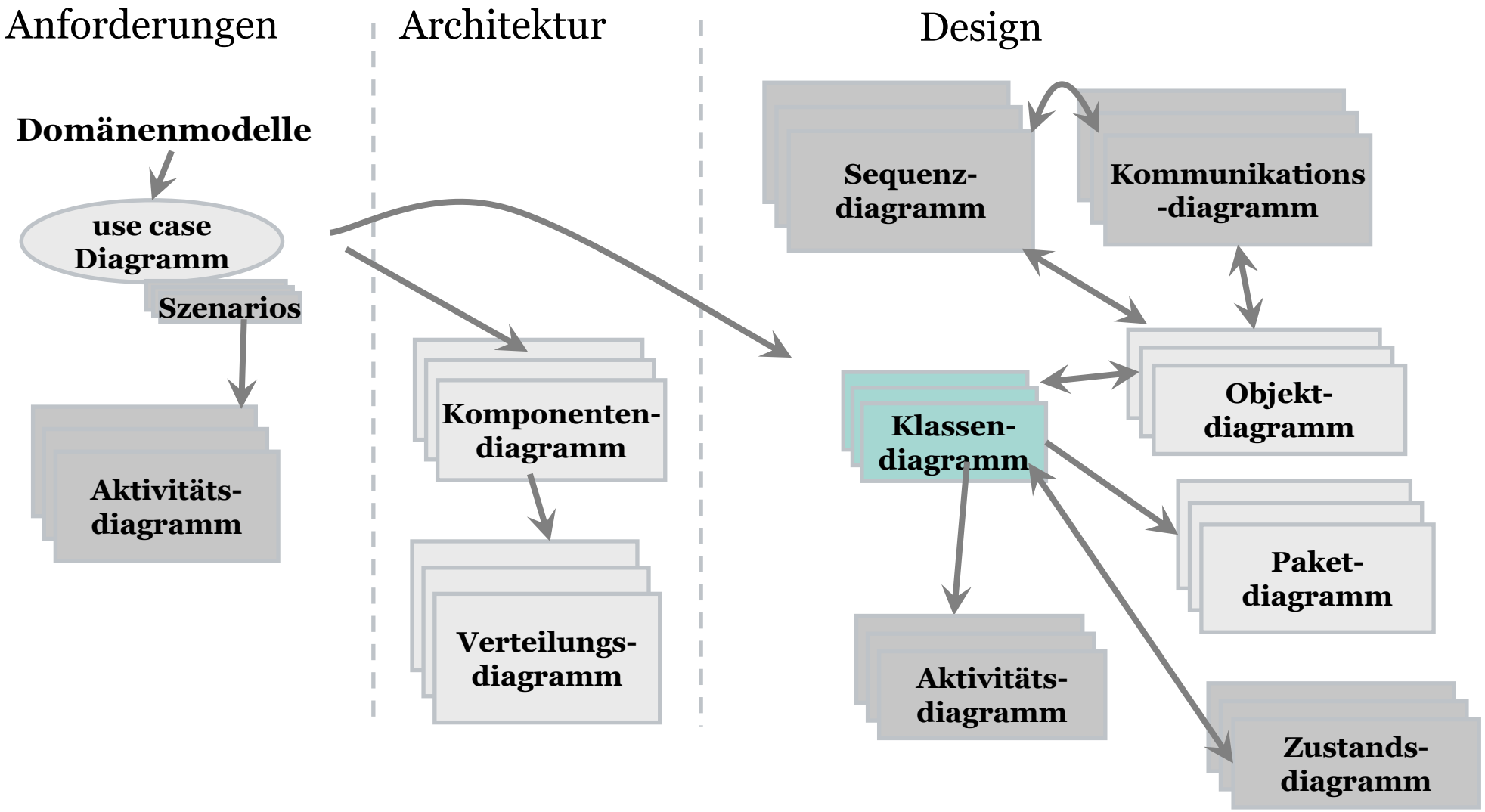
Strukturdiagramme (statisch)

- Klassendiagramm
- Montagediagramm
- Komponentendiagramm
- Verteilungsdiagramm
- Objektdiagramm
- Paketdiagramm
- Profildiagramm

Verhaltensdiagramme (dynamisch)

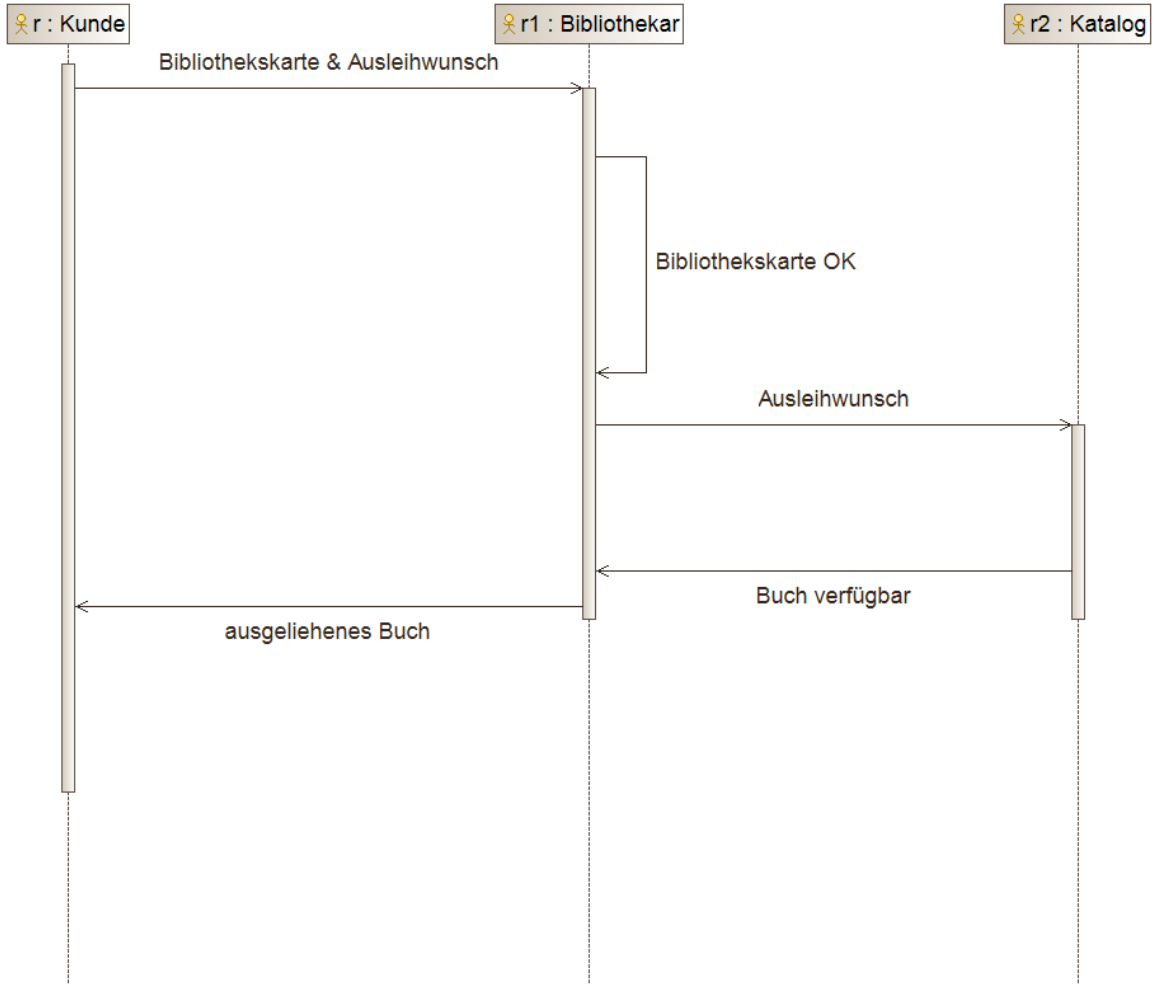
- Aktivitätsdiagramm
- Use-case Diagramm
- Interaktionsdiagramm
- Kommunikationsdiagramm
- Sequenzdiagramm
- Zeitverlaufdiagramm
- Zustandsdiagramm

Unified Modelling Language



Sequenzdiagramme

- Konstruktive Spezifikationsmethode
- Beschreibt Interaktion/Nachrichten zwischen Objekten
- Fokus auf Sequenz der Nachrichten



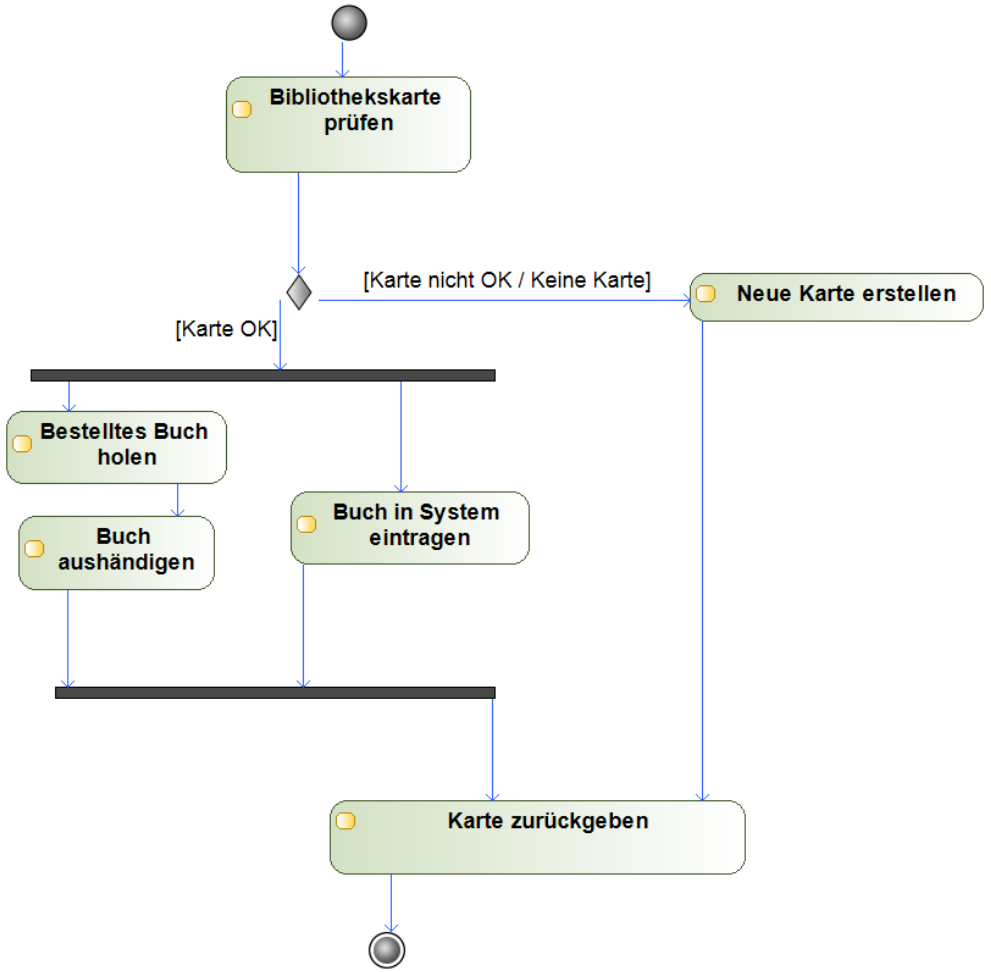
Kommunikationsdiagramme

Semantisch äquivalent zu Sequenzdiagramme

- Fokus auf Objektinteraktion

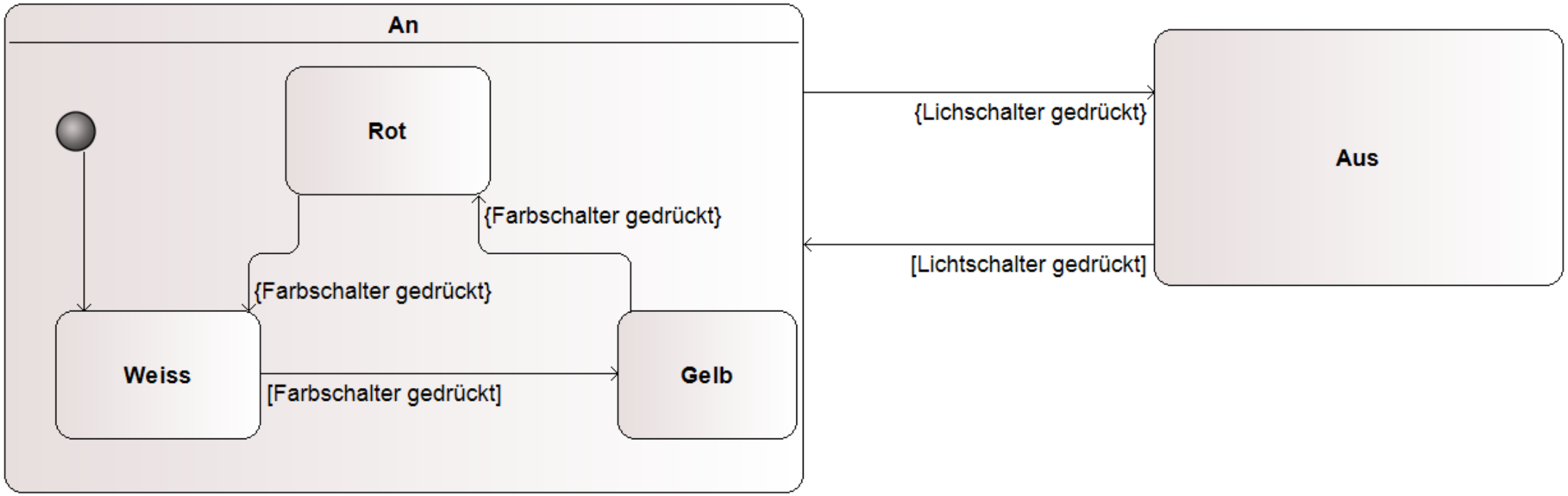
Aktivitätsdiagramme

- Konstruktive Spezifikationsmethode
- Modelliert Ablauf von Anwendungsfall
- Sequentielle und parallelen Aktivitäten möglich
    - Semantik basierend auf Petri Netzen



UML Zustandsdiagramme

Basierend auf Theorie der endlichen Automaten



UML für Spezifikationen

Use-case Diagramm	Finden von top-level Funktionen. Start vom Projekt.
Klassendiagramm	Anfangen mit wichtigsten Entitäten. Detaillieren während dem ganzen Projekt.
Sequenzdiagramm	Wichtige Szenarien dokumentieren. (Fokus: zeitliche Abfolge, Interaktion zwischen Objekten)
Aktivitätsdiagramm	Wichtige Szenarien dokumentieren (Fokus: Ablauf einzelner Aktivitäten)
Zustandsdiagramm	Spezifiziert Verhalten von Instanzen. Wird erst spät im Projekt eingesetzt.