
Bachelor of Science (BSc) in Informatik
Modul Advanced Software Engineering 1 (ASE1)

LE 05 - Requirements Engineering

6 Anforderungen modellbasiert dokumentieren

Institut für Angewandte Informationstechnologie (InIT)
Walter Eich (eicw) / Matthias Bachmann (bacn)

<https://www.zhaw.ch/de/engineering/institute-zentren/init/>

Agenda

6 Anforderungen modellbasiert dokumentieren

6.1 Der Modellbegriff

6.2 Zielmodelle

6.3 Use Cases

6.4 Drei Perspektiven auf die Anforderungen

6.5 Anforderungsmodellierung in der Strukturperspektive

6.6 Anforderungsmodellierung in der Funktionsperspektive

6.7 Anforderungsmodellierung in der Verhaltensperspektive

6.8 Wrap-up

Lernziele (1/2)

- LZ 6.1.1 Den **Modellbegriff** und die **Eigenschaften von Modellen** kennen
- LZ 6.1.2 **Definitionselemente** einer **konzeptuellen Modellierungssprache** kennen
- LZ 6.1.3 Die **Vorteile von Anforderungsmodellen** kennen
- LZ 6.2.1 Die **Bedeutung von Zielen** im Requirements Engineering kennen
- LZ 6.2.2 Die zwei Arten der **Zieldekomposition** kennen
- LZ 6.2.3 Die Modellierung von Zielbeziehungen in Und-Oder-Bäumen können und anwenden
- LZ 6.3.1 Die **Modellierung** von **Use-Case-Diagrammen** können und anwenden
- LZ 6.3.2 Die **Spezifikation** von **Use-Cases** können und anwenden
- LZ 6.4.1 Die **drei Perspektiven** auf Anforderungen kennen

Lernziele (2/2)

LZ 6.5.1 Die Bedeutung der **Strukturperspektive** auf Anforderungen kennen

LZ 6.5.2 **Entity-Relationship-Diagramme** und **UML-Klassendiagramme** können und anwenden

LZ 6.6.1 Den Fokus der **Funktionsperspektive** auf Anforderungen kennen

LZ 6.6.2 **Datenflussdiagramme** und **UML-Aktivitätsdiagramme** können und anwenden

LZ 6.7.1 Den Fokus der **Verhaltensperspektive** auf Anforderungen kennen

LZ 6.7.2 **UML-Zustandsdiagramme** können und anwenden

Einführung

- Im Rahmen der modellbasierten Dokumentation von Anforderungen werden **drei Ausprägungen von Anforderungen** dokumentiert, die im RE ergänzend zueinander eingesetzt werden:
 - **Ziele** beschreiben Intentionen von Stakeholdern oder Stakeholdergruppen.
 - **Use Cases und Szenarien** dokumentieren beispielshafte Abläufe der Systemnutzung. Szenarien werden in Use Cases gruppiert oder davon User Stories abgeleitet.
 - **Systemanforderungen** (allgemein als Anforderungen bezeichnet) beschreiben detaillierte Funktionalitäten und Qualitäten, die das zu entwickelnde System umsetzen soll.
- In der Praxis werden Anforderungen häufig in natürlicher Sprache formuliert. Es ist allerdings zu beobachten, dass vermehrt Anforderungen **zusätzlich** in Form von **Modellen** dokumentiert werden.

Einordnung in das Requirements Abstraction Model (RAM)

Abstraktionsebenen

Begrifflichkeit

Techniken

Organizational Strategies

Product Strategies

RAM - Abstraction Levels

Product Level (goal)

Feature Level (features)

Function Level (functions/actions)

Component Level (details- consists of)

Ziele

Features

**Szenarien,
Systemanforderungen**

Zielbäume

Featureliste

Use Cases,
Satzschablonen
UML-Diagramme für
Struktur-, Funktions- und
Verhaltensperspektive

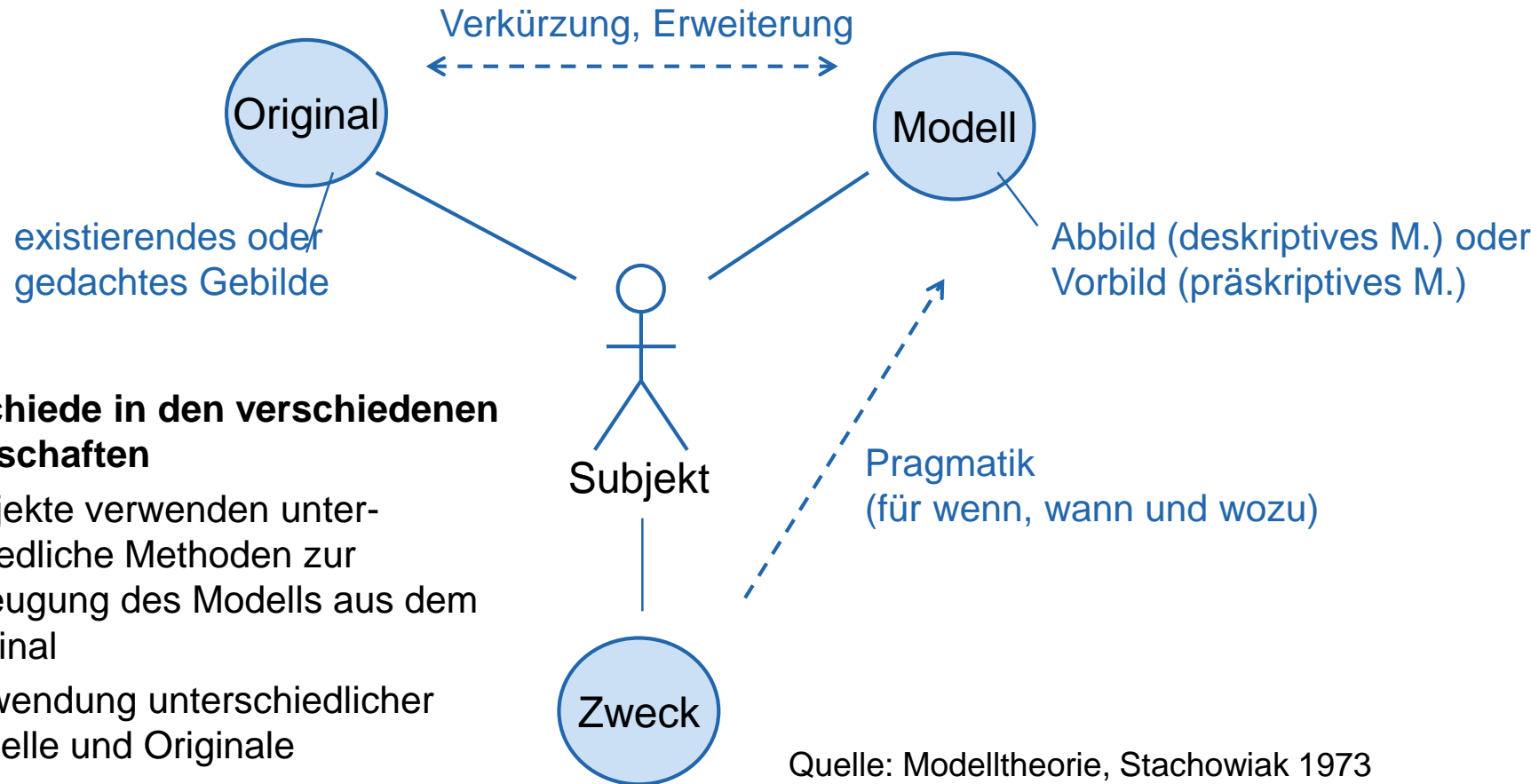
6.1 Der Modellbegriff

- Die Verwendung von Modellen erleichtert es, Informationen über einen Sachverhalt und deren Zusammenhänge gezielt zu verstehen, diese schneller zu erfassen und eindeutig zu dokumentieren.

Definition: Modell

Ein Modell ist ein abstrahierendes Abbild einer existierenden Realität oder Vorbild für eine zu schaffende Realität

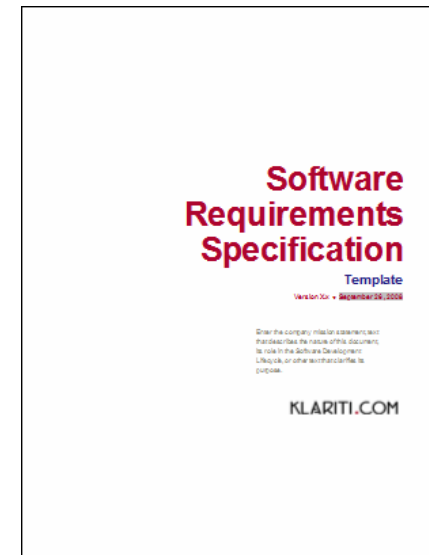
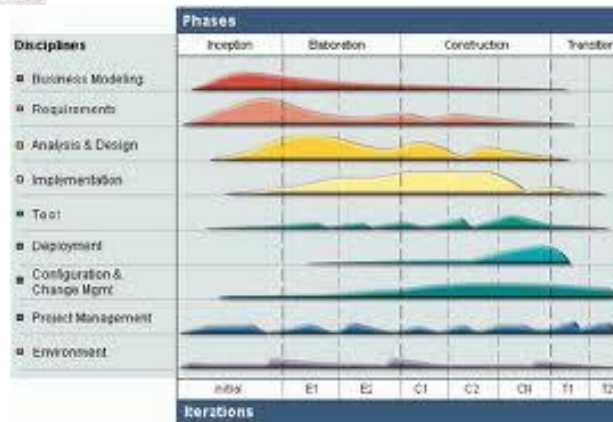
6.1.1 Eigenschaften von Modellen



Unterschiede in den verschiedenen Wissenschaften

- Subjekte verwenden unterschiedliche Methoden zur Erzeugung des Modells aus dem Original
- Verwendung unterschiedlicher Modelle und Originale
- Unterschiedlicher Zweck der Modellbildung

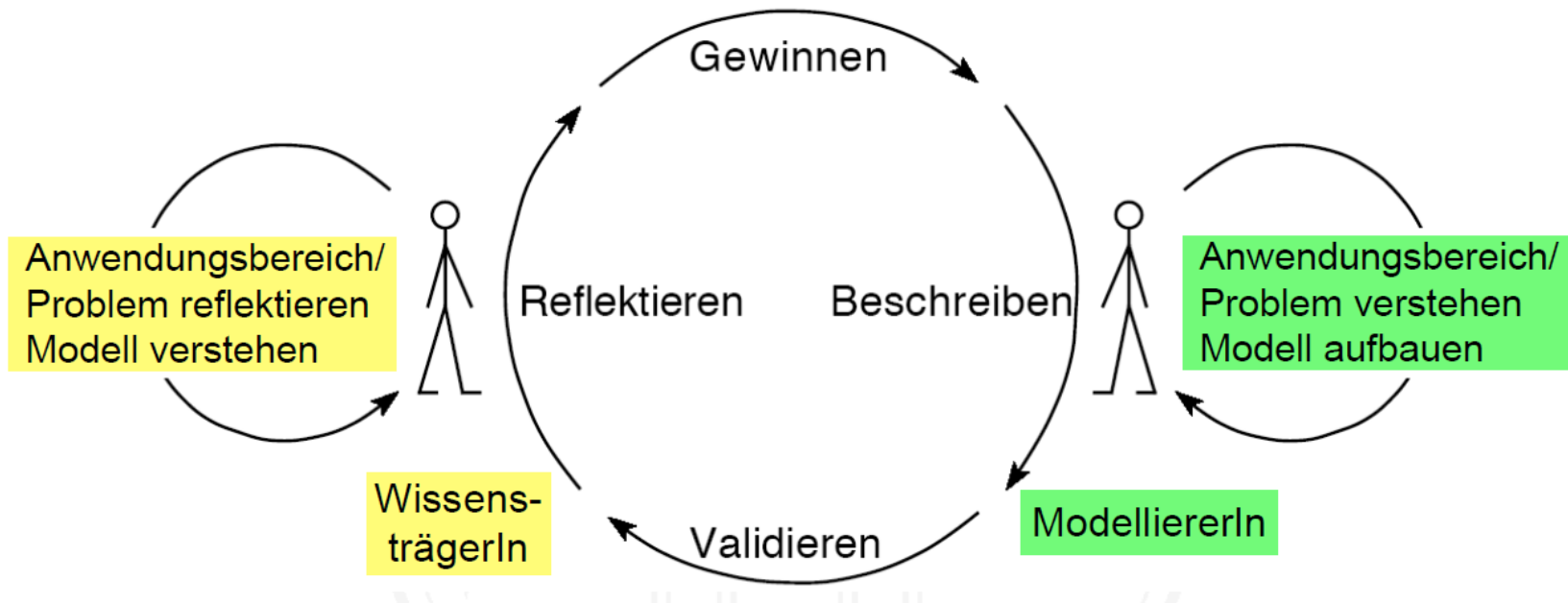
Was für Eigenschaften haben diese Modelle?



Wozu Modelle?

- **Verstehen** eines Gebildes (hier zu entwickelnde oder existierende Software)
- **Kommunizieren** über ein Gebilde
- Gedankliches **Hilfsmittel zum Gestalten, Bewerten** oder **Kritisieren** eines geplanten Gebildes oder von Varianten davon
- **Spezifikation** von Anforderungen an ein geplantes Gebilde
- Durchführung von **Experimenten**, die am Original nicht durchgeführt werden sollen, können oder dürfen
- Aufstellen / Prüfen von **Hypothesen** über beobachtete oder postulierte Phänomene

Prinzipschema der Modellbildung



Quelle: Vorlesungsskript Software Engineering,
Glinz 2006

- Modellbildung ist ein **iterativer Prozess**
- Modellieren bedeutet **immer auch Reflektieren** über das Original – unabhängig ob dieses bereits existiert oder erst zu schaffen ist
- Modellbildung ist auch ein **Verstehens- und Konsensbildungsprozess**

6.1.2 Konzeptuelle Modellierungssprachen

- Im RE werden häufig **konzeptuelle Modelle** eingesetzt, die in der Regel die abzubildende Realität durch eine Menge grafischer Elemente modellieren.
- Zur Modellierung konzeptueller Modelle werden **konzeptuelle Modellierungssprachen** eingesetzt, die über deren **Syntax** (Modellierungselemente und deren gültige Kombinationen) und **Semantik** (Bedeutung der Modellierungselemente) definiert werden.
- **Konzeptuelle Modelle können hinsichtlich ihres Formalisierungsgrades in informale, semiformale und formale Modellierungssprachen eingeteilt werden.**
- Der Formalisierungsgrad ist abhängig vom Umfang, in dem Syntax und Semantik der Sprache formal (z.B. über ein mathematisches Kalkül) definiert sind.

6.1.3 Anforderungsmodelle

- **Konzeptuelle Modelle**, die Anforderungen eines Systems dokumentieren, werden als **Anforderungsmodelle** bezeichnet
- Zur Konstruktion von Anforderungsmodellen wird oft **UML** eingesetzt
- **UML** besteht aus einer Menge von zum Teil **komplementären Modellierungssprachen**, die speziell im RE eingesetzt werden um die Anforderungen eines Systems aus **verschiedenen Perspektiven** zu modellieren
- **Unterschied konzeptueller Modelle in der Systementwicklung** der Verwendung von konzeptuellen Modellen in der Anforderungsdokumentation:
 - Systementwicklung: es werden Lösungsaspekte dokumentiert
 - RE: es werden Anforderungen aus verschiedenen Perspektiven modelliert

Modellierungssprachen im RE für Ziele und Szenarien

- **Ziele**
 - Und-Oder-Graphen
- **Szenarien**
 - Personas und Szenarien
 - Storyboards, UI-Sketches
 - Use Case Diagramme (und Spezifikationen) ergänzt durch:
 - UML-Aktivitätsdiagramme
 - UML-Sequenzdiagramme
 - UML-Zustandsdiagramme

} aus dem Usability Engineering

Modellierungssprachen im RE für Systemanforderungen

- **Funktionsperspektive**
 - UML-Aktivitätsdiagramme
 - UML-Sequenzdiagramme
 - Datenflussdiagramme (DFD)
- **Strukturperspektive**
 - UML-Klassendiagramme
 - Entity-Relationship-Diagramme (ERD)
- **Verhaltensperspektive**
 - UML-Zustandsdiagramme
 - Statecharts

Die drei Perspektiven sind nicht disjunkt. In der objektorientierten Modellierung mit UML werden diese drei Sichten integriert.

6.1.4 Vorteile von Anforderungsmodellen

- Die Dokumentation von Anforderungen in Form **konzeptueller Modelle** bietet im Vergleich zur natürlichsprachigen Dokumentation von Anforderungen unter anderem folgende **Vorteile**:
 - **Bildhafte** dargestellte Information kann **schneller erfasst** und memorisiert werden
 - Mit **Anforderungsmodellen** kann gezielt **eine Perspektive** auf Anforderungen modelliert werden.
 - Durch die Definition der Modellierungssprache für den **jeweiligen Verwendungszweck** können bereits zweckmässige **Abstraktionen der Realität** festgelegt werden.

Vorteile von Modellen im RE

- Ein Bild sagt mehr als 1000 Worte!
- Bildung diskreter Perspektiven (z.B. Domänenmodell aus der Struktur- bzw. Datenperspektive)
- Beispiel:

Struktur-/Datenperspektive

Verhaltensperspektive

Funktionsperspektive

Stellt ein an der Eingangstür befestigter Glasbruchsensor fest, dass die Eingangstür beschädigt wurde, soll das System in den Alarmzustand übergehen und nach spätestens 2 Sekunden den Sicherheitsdienst benachrichtigen.
- Konzeptuelle Modellierungssprachen bieten die Möglichkeit zur Abstraktion (z.B. durch Klassifizierung, Komposition, Generalisierung, Benutzung).

6.1.5 Kombiniertes Einsatz - Modelle und Sprache

- Die **Kombination** von natürlicher Sprache und **Anforderungsmodelle** kombiniert die Vorteile beider Dokumentationsarten.
- Natürlich **sprachliche Anforderungen** können **durch Modelle** **zusammengefasst** werden
- **Modelle** können verwendet werden um natürlich **sprachliche Anforderungen** zu **detaillieren**

Agenda

6 Anforderungen modellbasiert dokumentieren

6.1 Der Modellbegriff

6.2 Zielmodelle

6.3 Use Cases

6.4 Drei Perspektiven auf die Anforderungen

6.5 Anforderungsmodellierung in der Strukturperspektive

6.6 Anforderungsmodellierung in der Funktionsperspektive

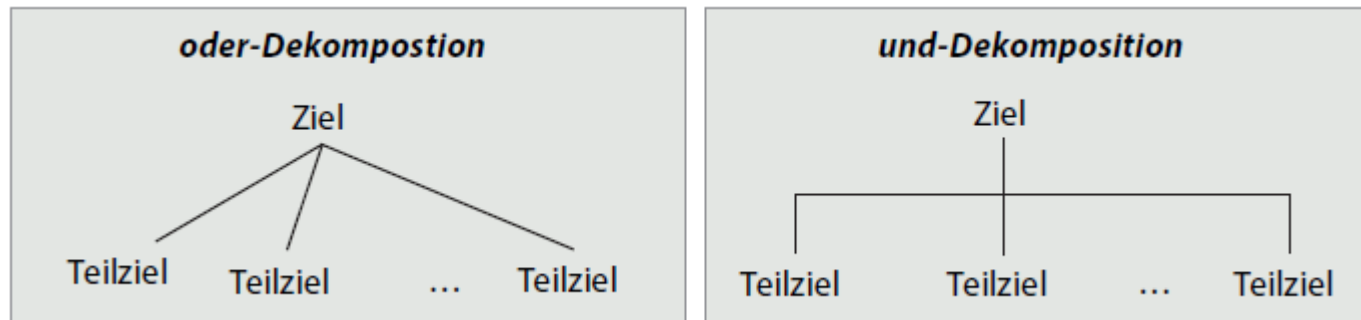
6.7 Anforderungsmodellierung in der Verhaltensperspektive

6.8 Wrap-up

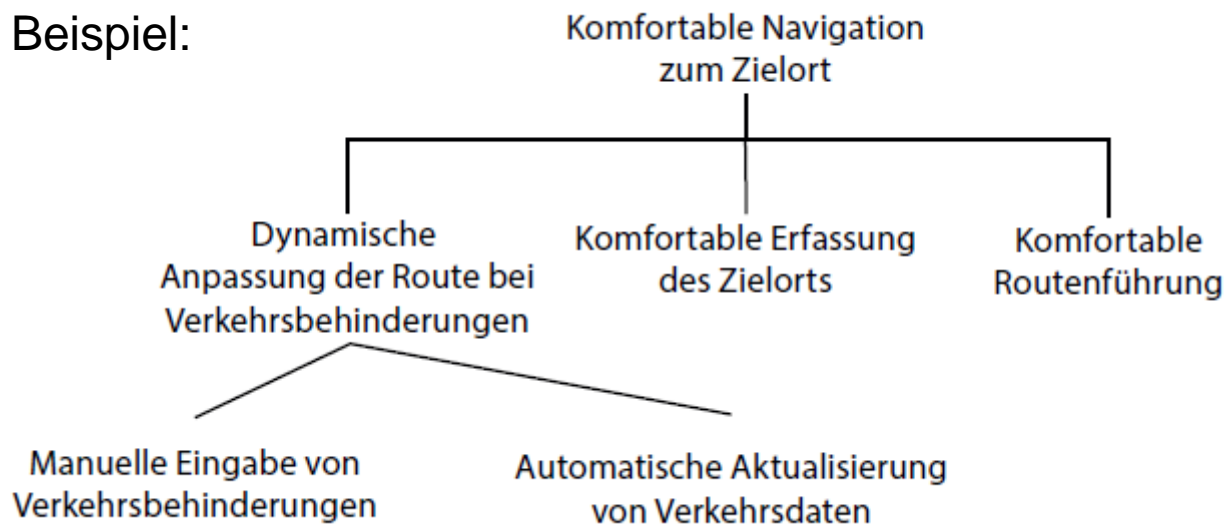
6.2 Zielmodelle

- Ein Ziel ist die intentionale Beschreibung eines von Stakeholdern gewünschten charakteristischen Merkmals des zu entwickelnden Systems bzw. des zugehörigen Entwicklungsprojekts.
- Ziele können sowohl natürlichsprachig als auch in Form von Modellen dokumentiert werden.
- Ein wesentlicher Bestandteil der Dokumentation von Zielen ist die Beschreibung von Verfeinerungsbeziehungen (Dekompositionsbeziehungen) zwischen einem übergeordneten und untergeordneten Zielen.

6.2.1 Zieldokumentation mit Und-Oder Bäumen



Beispiel:



Agenda

6 Anforderungen modellbasiert dokumentieren

6.1 Der Modellbegriff

6.2 Zielmodelle

6.3 Use Cases

6.4 Drei Perspektiven auf die Anforderungen

6.5 Anforderungsmodellierung in der Strukturperspektive

6.6 Anforderungsmodellierung in der Funktionsperspektive

6.7 Anforderungsmodellierung in der Verhaltensperspektive

6.8 Wrap-up

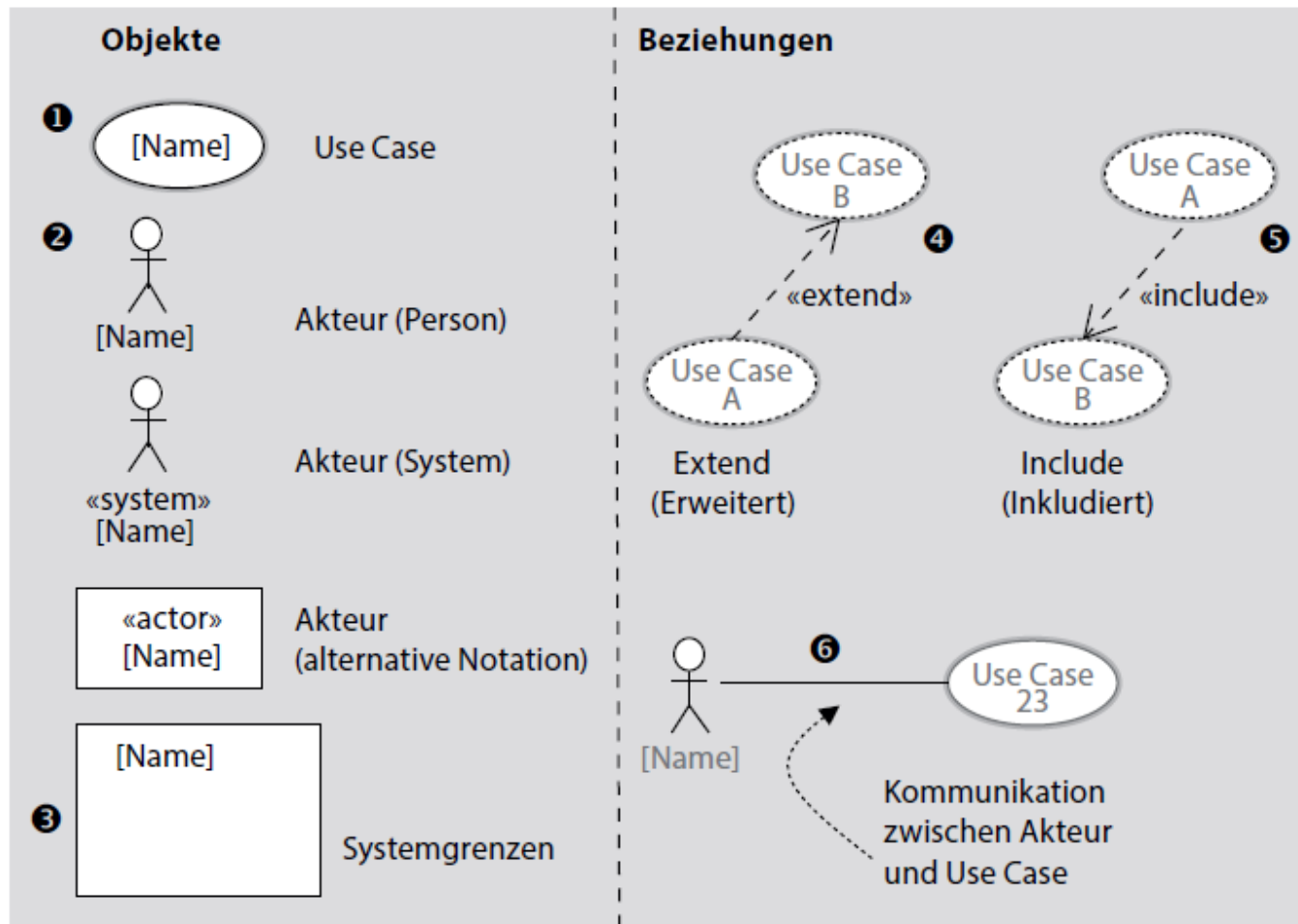
6.3 Use Cases

- Use Cases dienen dazu, die **Funktionalität** eines geplanten oder existierenden Systems aus einer **Nutzungssicht** auf das System untersuchen und dokumentieren zu können.
- Der Use-Case-Ansatz basiert auf zwei sich ergänzenden Dokumentationstechniken:
 - Use-Case-Diagramme
 - Use-Case-Spezifikationen

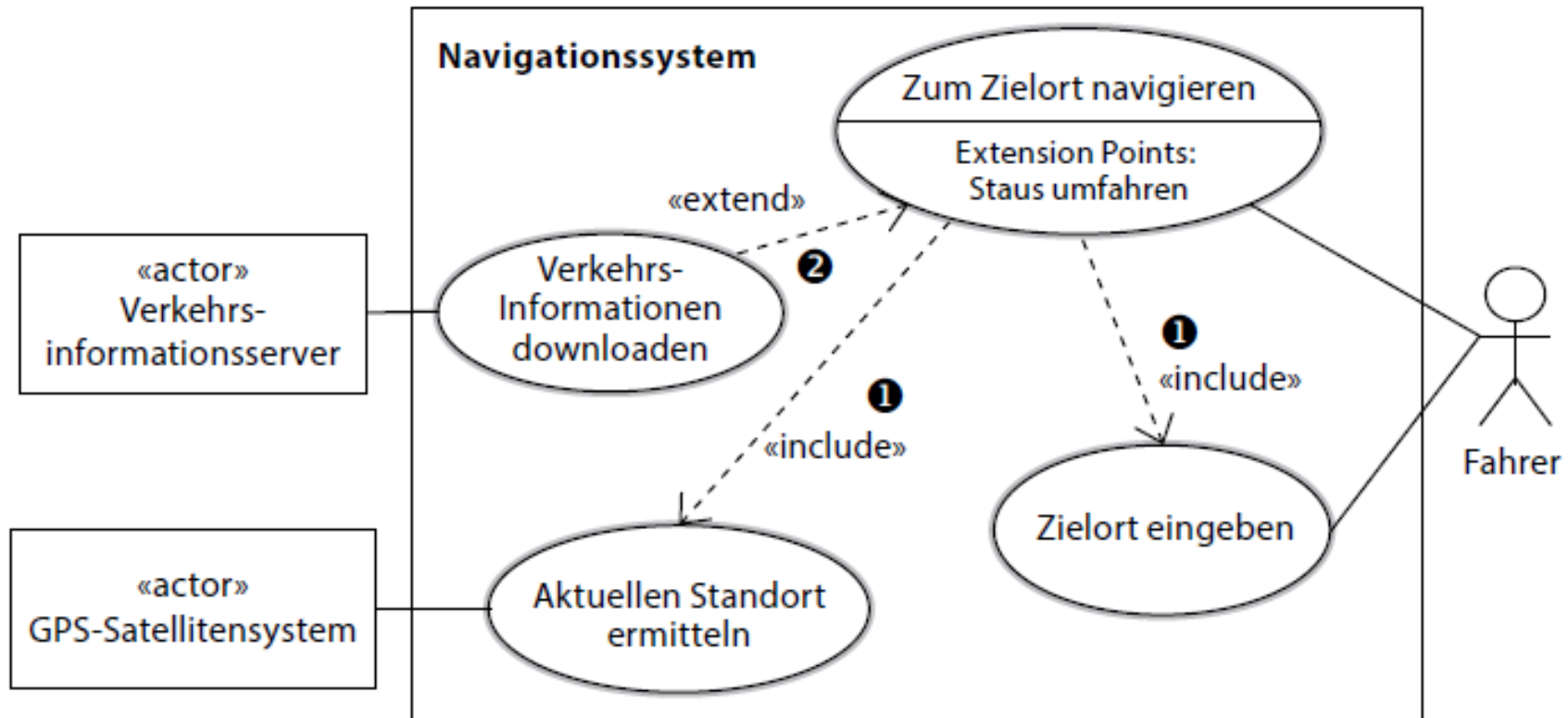
6.3.1 UML-Use-Case-Diagramme

- Use-Case-Diagramme sind leicht verständliche Modelle, welche die aus einer **Nutzungssicht** notwendigen **Funktionen** des betrachteten Systems, deren Beziehungen untereinander sowie den **Kontext** des Systems dokumentieren.
- Typische Modellierungselemente von Use-Case-Diagrammen sind:
 - **Akteure** (Personen oder andere Systeme) im Systemkontext
 - die **Systemgrenze**
 - **Use Cases**
 - verschiedene Typen von Beziehungen zwischen diesen Modellierungselementen.
 - **Extend-Beziehung**
 - **Include-Beziehung**
 - Beziehung zwischen Akteuren und Use Cases

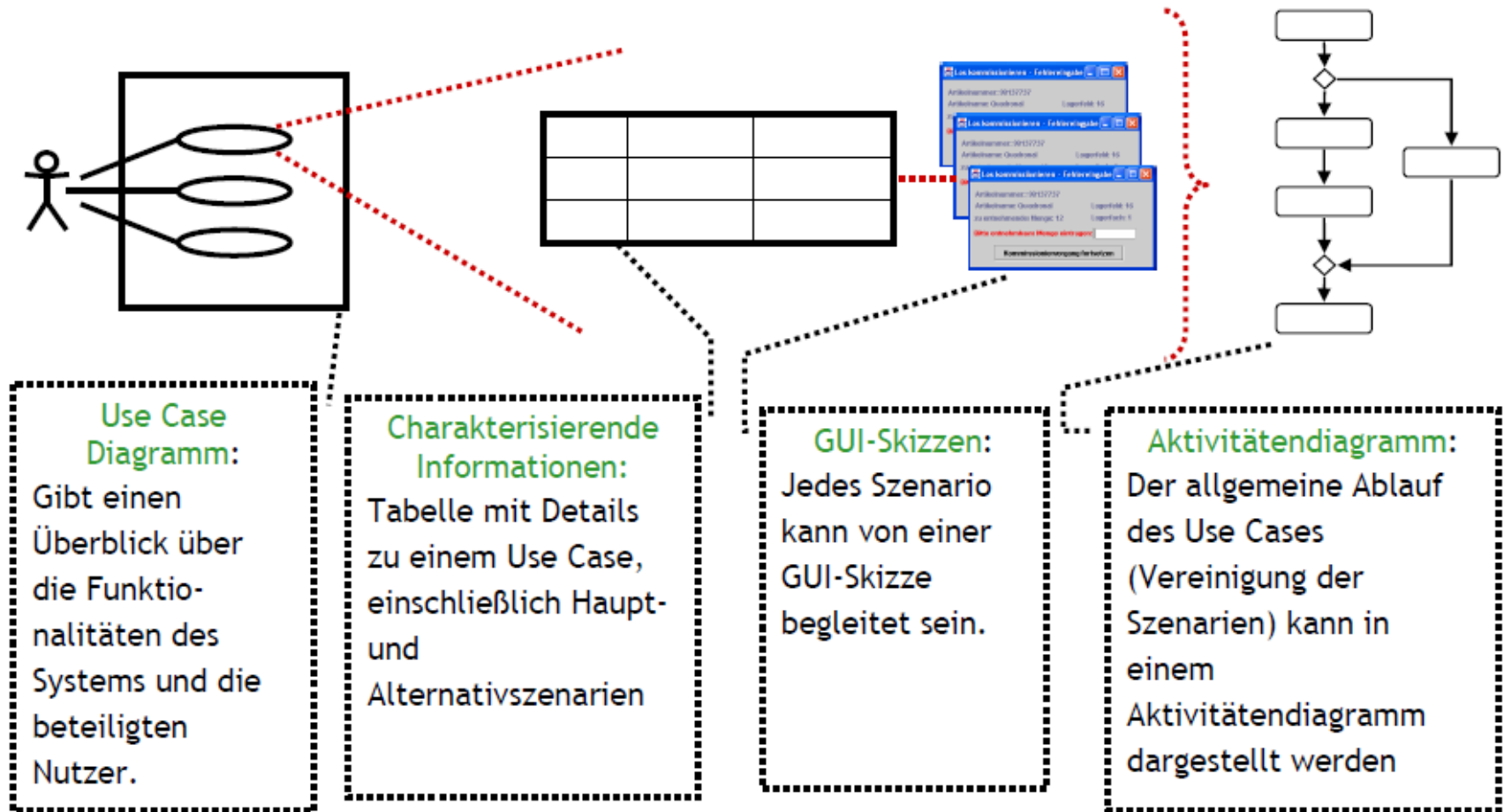
Modellelemente Use Case



Beispiel für UML Use Case Diagramme



Beschreibung von Anwendungsfällen



6.3.2 Use Case Spezifikationen

- **Use-Case-Spezifikationen** ergänzen die überblicksartigen Use-Case-Diagramme durch eine genaue Spezifikation der wesentlichen Eigenschaften einzelner Use Cases.
- Hierzu wird in der Regel für jeden relevanten Use Case separat eine vorgegebene Schablone ausgefüllt.

6.3.2 Use-Case-Spezifikationen (1/2)

Schablone zur Spezifikation eines Use Case		
Nr.	Abschnitt	Inhalt / Erläuterung
1	Bezeichner	Eindeutiger Bezeichner des Use Case
2	Name	Eindeutiger Name für den Use Case
3	Autoren	Namen der Autoren, die an dieser Use-Case-Beschreibung mitgearbeitet haben
4	Priorität	Wichtigkeit des Use Case gemäß der verwendeten Priorisierungstechnik
5	Kritikalität	Kritikalität des Use Case, z.B. hinsichtlich des Schadensausmaßes bei Fehlverhalten des Use Case
6	Quelle	Bezeichnung der Quelle ([Stakeholder Dokument System]), von der der Use Case stammt
7	Verantwortlicher	Der für diesen Use Case verantwortliche Stakeholder
8	Beschreibung	Komprimierte Beschreibung des Use Case
9	Auslösendes Ereignis	Angabe des Ereignisses, das den Use Case auslöst

6.3.2 Use-Case-Spezifikationen (2/2)

Schablone zur Spezifikation eines Use Case		
Nr.	Abschnitt	Inhalt/Erläuterung
10	Akteure	Auflistung der Akteure, die mit dem Use Case in Beziehung stehen
11	Vorbedingung	Eine Liste notwendiger Voraussetzungen, die erfüllt sein müssen, bevor die Ausführung des Use Case beginnen kann
12	Nachbedingung	Eine Liste von Zuständen, in denen sich das System unmittelbar nach der Ausführung des Hauptszenarios befindet.
13	Ergebnis	Beschreibung der Ausgaben, die während der Ausführung des Use Case erzeugt werden
14	Hauptszenario	Beschreibung des Hauptszenarios eines Use Case
15	Alternativ-szenarien	Beschreibung von Alternativszenarien des Use Case oder lediglich Angabe der auslösenden Ereignisse. Hier gelten oftmals andere Nachbedingungen als in (12) enthalten.
16	Ausnahme-szenarien	Beschreibung von Ausnahmeszenarien des Use Case oder lediglich Angabe der auslösenden Ereignisse. Hier gelten oftmals andere Nachbedingungen als in (12) enthalten.
17	Qualitäten	Querbezüge zu Qualitätsanforderungen

Beispiel Use-Case-Spezifikation (1/2)

Abschnitt	Inhalt
Bezeichner	UC-12-37
Name	Zum Zielort navigieren
Autoren	Bernd Schmitz, Klaus Müller
Priorität	Wichtigkeit für Systemerfolg »hoch« Technologisches Risiko »hoch«
Kritikalität	Hoch
Quelle	C. Schulz (Domänenexperte für Navigationssysteme)
Verantwortlicher	B. Schmitz
Kurzbeschreibung	Der Fahrer des Fahrzeugs gibt den Zielort ein. Das Navigationssystem leitet den Fahrer zum gewünschten Zielort.
Auslösendes Ereignis	Fahrer möchte zu einem Ziel navigieren.
Akteure	Fahrer, Informationsserver, GPS-Empfänger
Vorbedingung	Das Navigationssystem ist eingeschaltet.
Nachbedingung	Der Fahrer hat sein Ziel erreicht.
Ergebnis	Wegführung zum Zielort

Beispiel Use-Case-Spezifikation (2/2)

Hauptszenario	<ol style="list-style-type: none"> 1. Das Navigationssystem erfragt den gewünschten Zielort. 2. Der Fahrer gibt den gewünschten Zielort ein. 3. Das Navigationssystem ermittelt den Zielort in seinen Karten. 4. Mithilfe des aktuellen Standorts und des gewünschten Zielorts berechnet das Navigationssystem eine Route. 5. Das Navigationssystem stellt die Liste der Wegpunkte zusammen. 6. Das Navigationssystem zeigt die Karte des aktuellen Standorts und die Navigation zum nächsten Wegpunkt an. 7. Wurde der letzte Wegpunkt erreicht, zeigt das Navigationssystem am Bildschirm an »Ziel erreicht«.
Alternativ-szenarien	<ol style="list-style-type: none"> 4a. Die Routenberechnung soll Verkehrsinformationen beachten und Staus umfahren. <ol style="list-style-type: none"> 4a1. Das Navigationssystem erfragt beim Informationsserver die aktuellen Verkehrsinformationen. 4a2. Das Navigationssystem ermittelt eine Route, die keine vom Stau betroffenen Wegstrecken enthält.
Ausnahme-szenarien	Auslösendes Ereignis: Das Navigationssystem empfängt kein GPS-Signal vom GPS-Empfänger.
Qualitäten	<p>→ QA.04 (Reaktionszeit auf Benutzereingaben)</p> <p>→ QA.15 (Bedienungskomfort)</p> <p>(QA = Qualitätsanforderungen)</p>

UI Sketching & Prototyping

- Parallel zur Erhebung von Use Cases ist es empfehlenswert, UI-Sketches und Prototypen zu erstellen.
- Damit können weitere Anforderungen erhoben, Anforderungen geklärt und evaluiert werden.
- Das Prototyping kann mit Papier oder speziellen Prototyping-Tools gemacht werden.
- Ein empfehlenswertes Tool ist www.balsamiq.com (eine Lizenz zum Ausprobieren ist auf OLAT in SW# 5 verfügbar).



Agenda

6 Anforderungen modellbasiert dokumentieren

6.1 Der Modellbegriff

6.2 Zielmodelle

6.3 Use Cases

6.4 Drei Perspektiven auf die Anforderungen

6.5 Anforderungsmodellierung in der Strukturperspektive

6.6 Anforderungsmodellierung in der Funktionsperspektive

6.7 Anforderungsmodellierung in der Verhaltensperspektive

6.8 Wrap-up

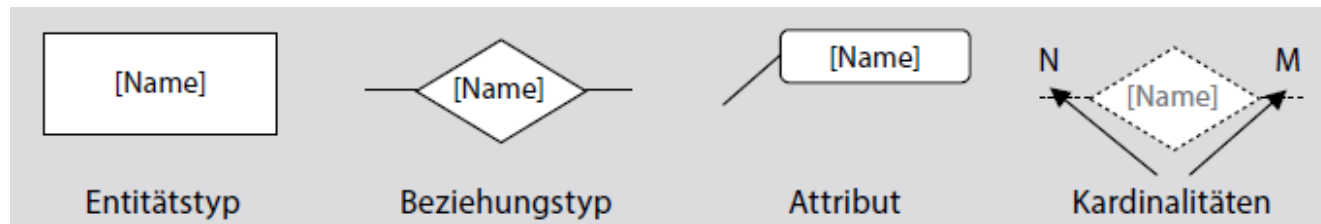
6.4 Die 3 Perspektiven auf die Anforderungen

Drei Modellierungsperspektiven:

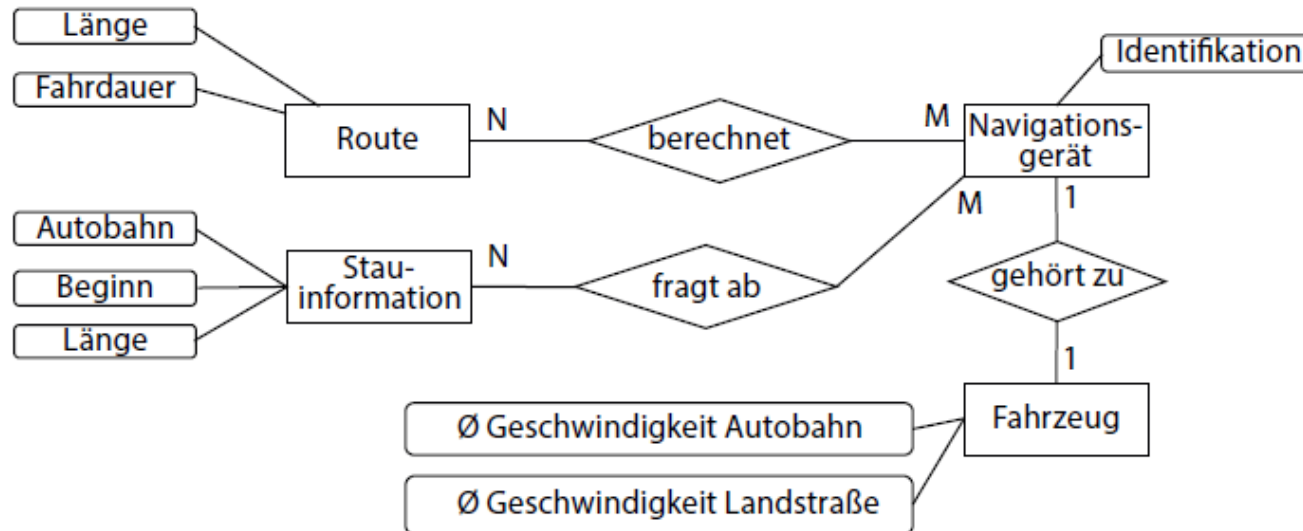
- **Strukturperspektive:** In dieser Perspektive werden die Struktur von **Ein- und Ausgabedaten** sowie die **statisch-strukturellen Aspekte von Nutzungs- und Abhängigkeitsbeziehungen** des Systems im Systemkontext dokumentiert.
- **Funktionsperspektive:** In dieser Perspektive wird dokumentiert, welche **Informationen** aus dem **Systemkontext** durch das zu entwickelnde System bzw. dessen **Funktionen manipuliert werden** und welche **Daten** vom **System** in den **Systemkontext** fließen
- **Verhaltensperspektive:** In dieser Perspektive wird das System und dessen Einbettung **zustandsorientiert** dokumentiert, indem z.B. die **Reaktion** des Systems auf **Ereignisse** im Systemkontext, Bedingungen eines **Zustandswechsels** sowie **Effekte** dokumentiert werden, die das System in der Umgebung erbringen soll.

6.5 Anforderungsmodellierung in der Strukturperspektive (1/2)

Mit Entity-Relationship-Diagramm (ERD):

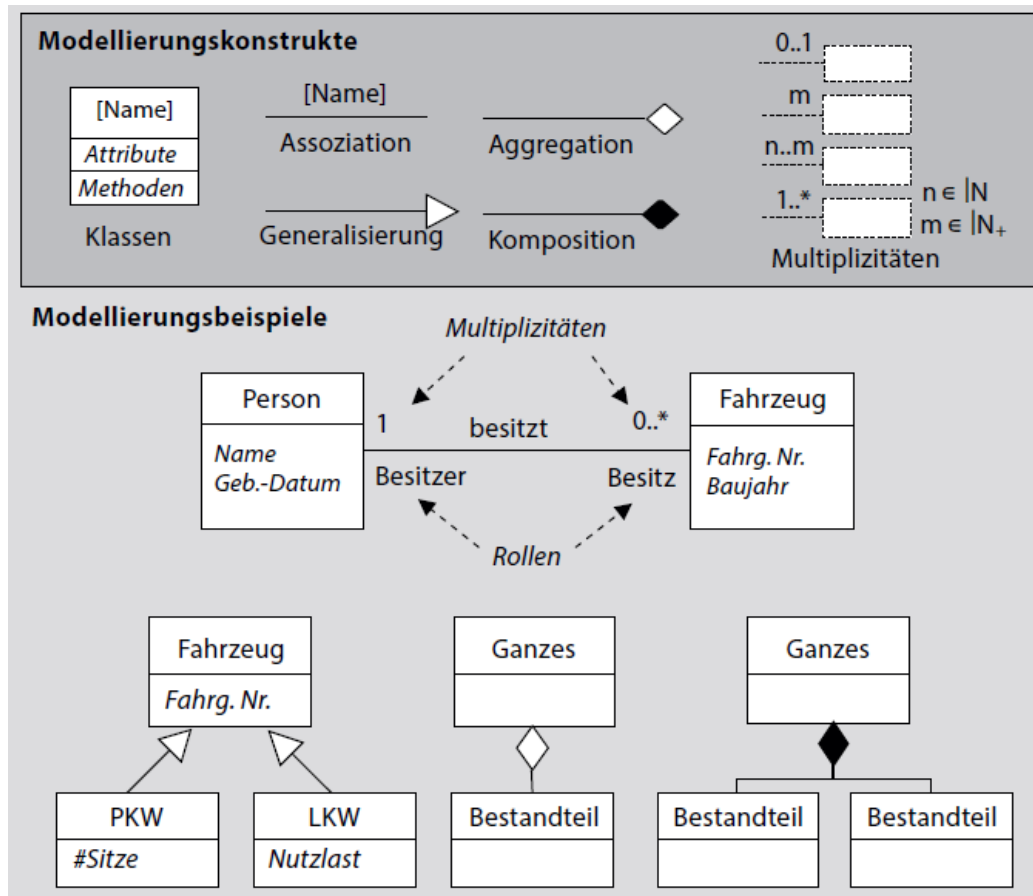


Beispiel:

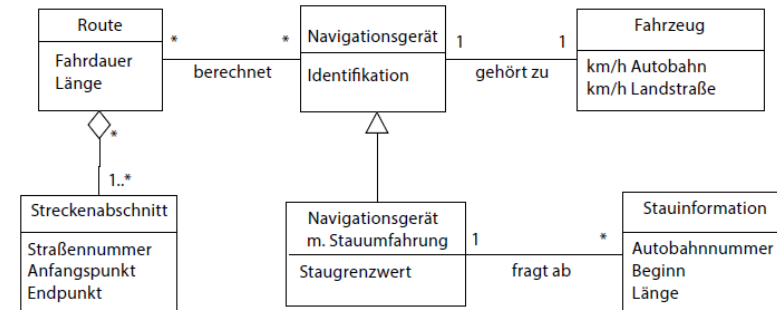


6.5 Anforderungsmodellierung in der Strukturperspektive (2/2)

Mit **UML-Klassendiagramm**-Diagramm:

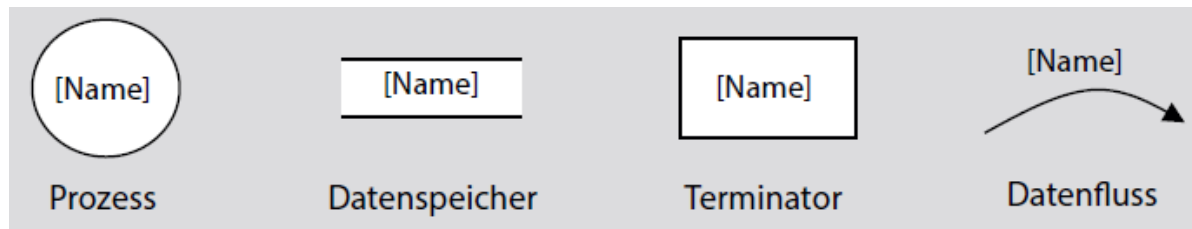


Beispiel:

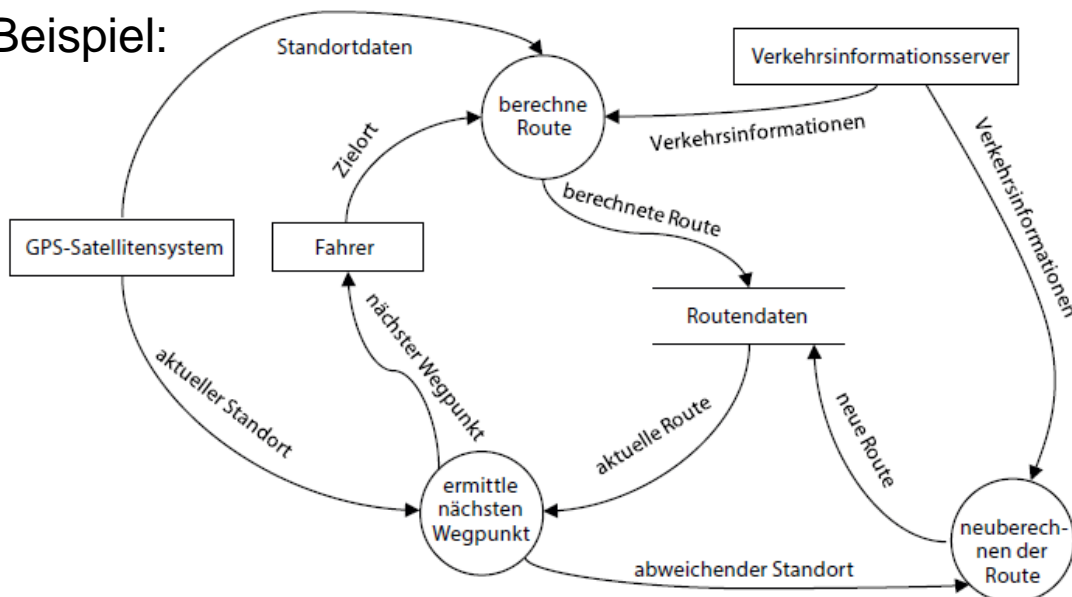


6.6 Anforderungsmodellierung in der Funktionssperspektive (1/2)

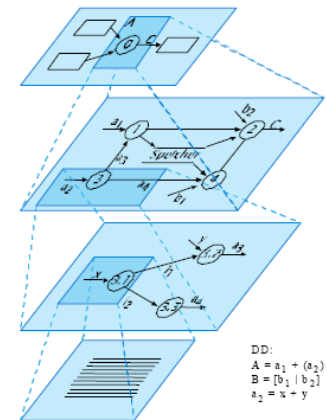
Mit **Datenflussdiagramm** (veraltet):



Beispiel:

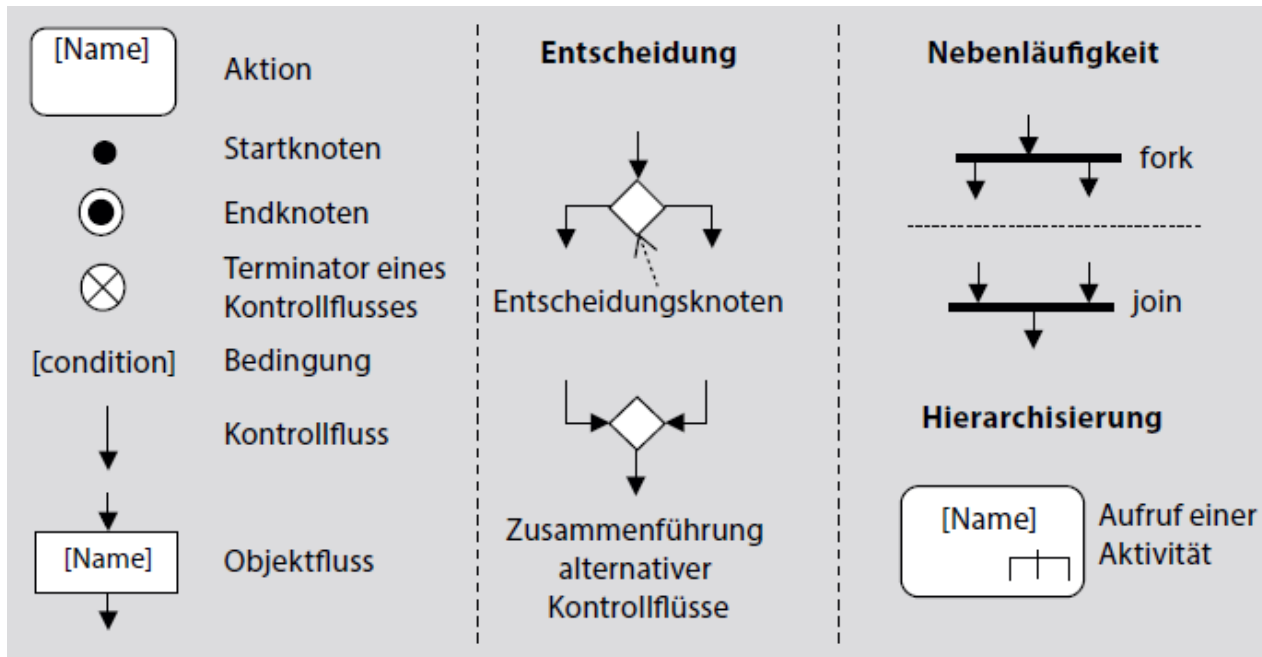


- Kontextdiagramm
- Diagramm 0
- Diagramm 3
- MiniSpec 3.1

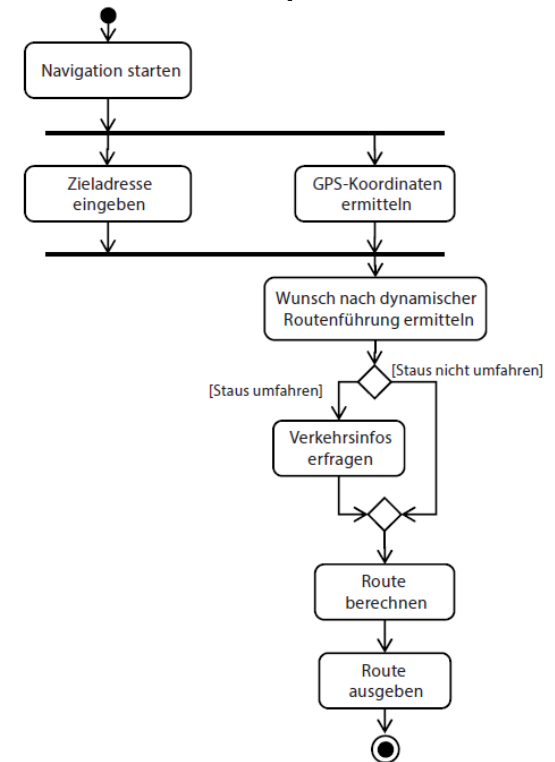


Funktionperspektive (2/2)

Mit UML-Aktivitätsdiagramm:

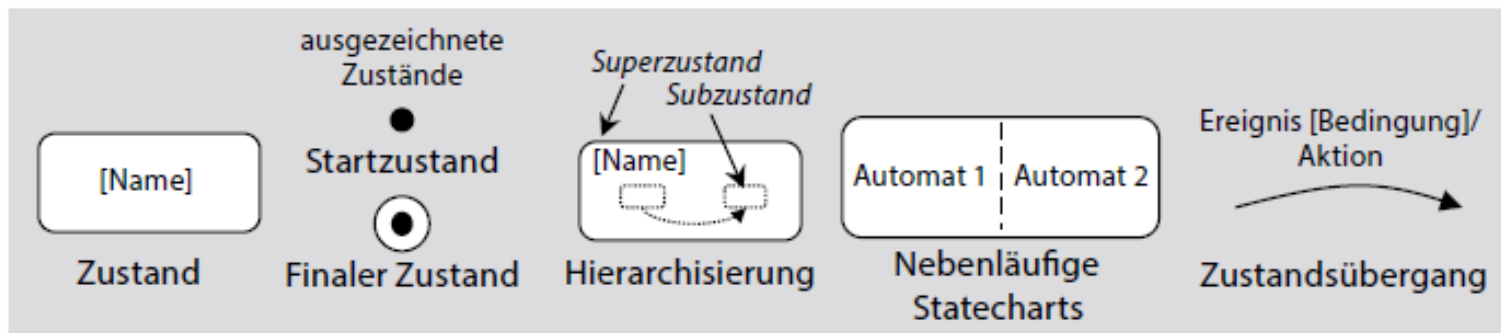


Beispiel:

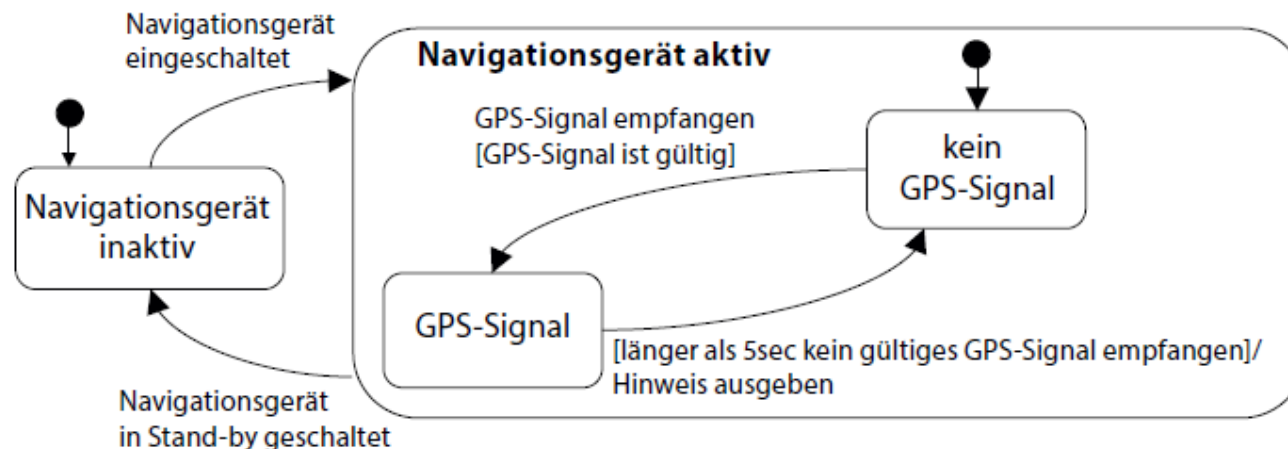


6.7 Anforderungsmodellierung in der Verhaltenperspektive (1/2)

Mit **Statechart** (veraltet):

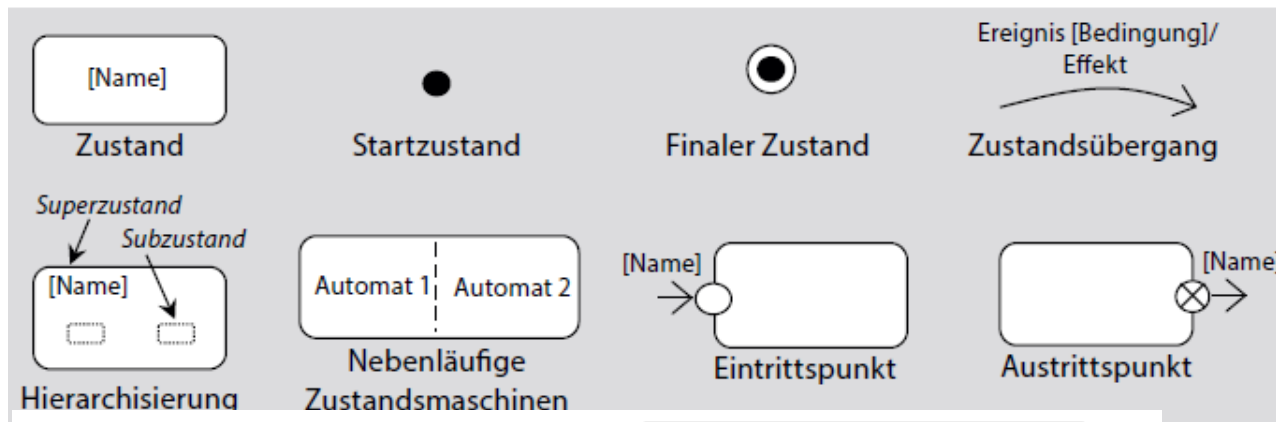


Beispiel:

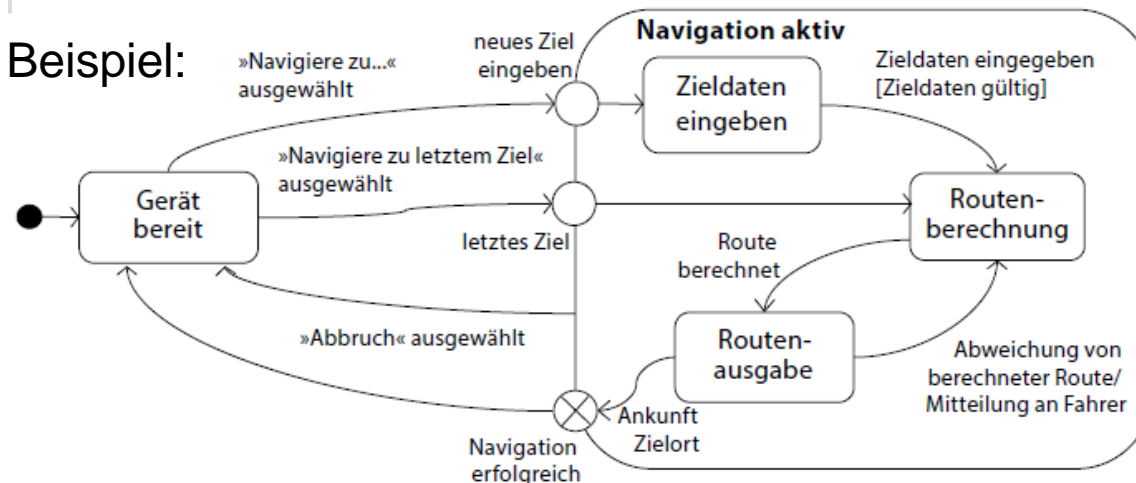


6.7 Anforderungsmodellierung in der Verhaltenperspektive (2/2)

Mit UML-Zustandsdiagramm:



Beispiel:



Agenda

6 Anforderungen modellbasiert dokumentieren

6.1 Der Modellbegriff

6.2 Zielmodelle

6.3 Use Cases

6.4 Drei Perspektiven auf die Anforderungen

6.5 Anforderungsmodellierung in der Strukturperspektive

6.6 Anforderungsmodellierung in der Funktionsperspektive

6.7 Anforderungsmodellierung in der Verhaltensperspektive

6.8 Wrap-up

Wrap-up

- Neben der natürlichsprachigen Dokumentation von Anforderungen können **Anforderungen** auch **in Form von Modellen** dokumentiert werden.
- Typischerweise werden die **natürlichsprachliche und modellbasierte Dokumentation** von Anforderungen heute oftmals **in Kombination eingesetzt**.
- Die **modellbasierte Dokumentation** von Anforderungen besitzt dabei u.a. den **Vorteil**, dass grafische (bildhafte) Beschreibungen von Sachverhalten in der Regel **schneller und besser verstanden** werden als natürlichsprachige Beschreibungen.
- Zu den häufig im Requirements Engineering eingesetzten konzeptuellen Modellen zählen **Zielmodelle** (z.B. in Form von Und-Oder-Bäumen) und **Use-Case-Diagramme** sowie **konzeptuelle Modelle** zur Dokumentation von Anforderungen aus den drei Perspektiven: **Strukturperspektive**, **Funktionsperspektive** und **Verhaltensperspektive**.