

# Softwareanforderungsanalyse

Vorgehen, Modellstruktur und Spezifikationsdokument -  
Ein Fazit

Burkhardt Renz

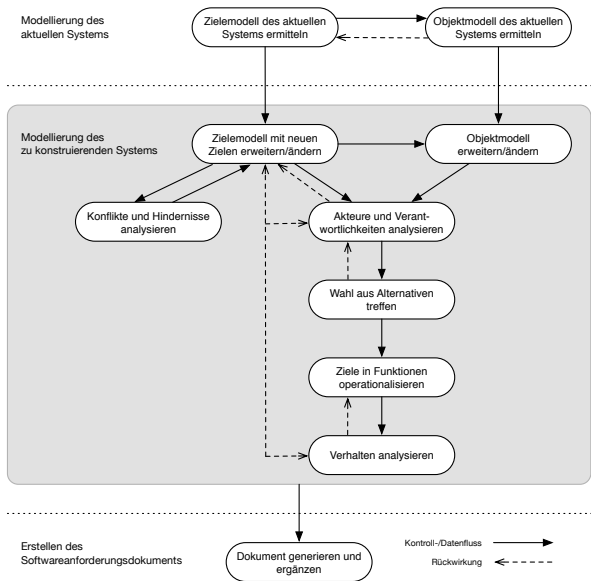
THM, Fachbereich MNI

Wintersemester 2018/19

# Übersicht

- Vorgehen
- Struktur des Modells – Metamodell
- Generierung des Anforderungsdokuments

# Vorgehen nach der KAOS-Methode



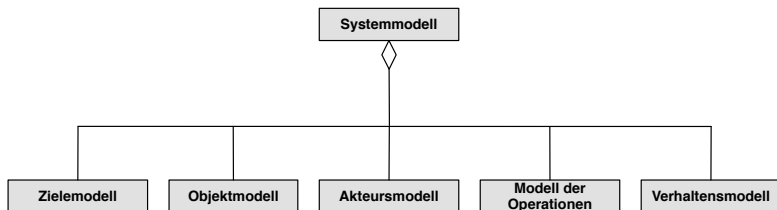
# Übersicht

- Vorgehen
- Struktur des Modells – Metamodell
- Generierung des Anforderungsdokuments

# Struktur des Modells – Metamodell

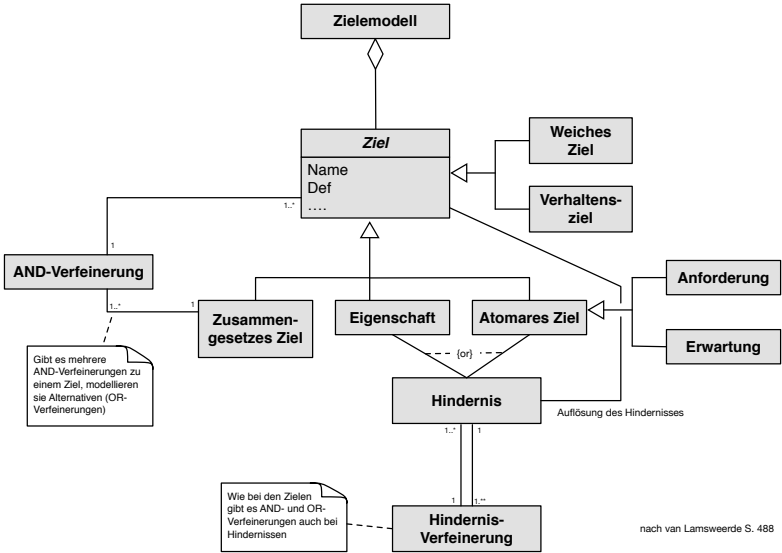
- **Instanzen** sind die konkreten Objekte  
z.B. das am 12.2. ausgeliehene Buch, das zweite Segment der Bahnstrecke nach der Station am Terminal 1, der geplante Termin am 20.12. . . .
- Ein **Modell** beschreibt Ziele, Konzepte und Objekttypen des Anwendungsgebiets  
z.B. Buchexemplar, Zug, Segment, Termin. . .
- Das **Metamodell** beschreibt die im Modell selbst vorkommenden Objekttypen und deren Zusammenhang und Constraints  
z.B. Ziel, Akteurstyp, Entitätstyp, Funktion. . .

# Die Teilmodelle



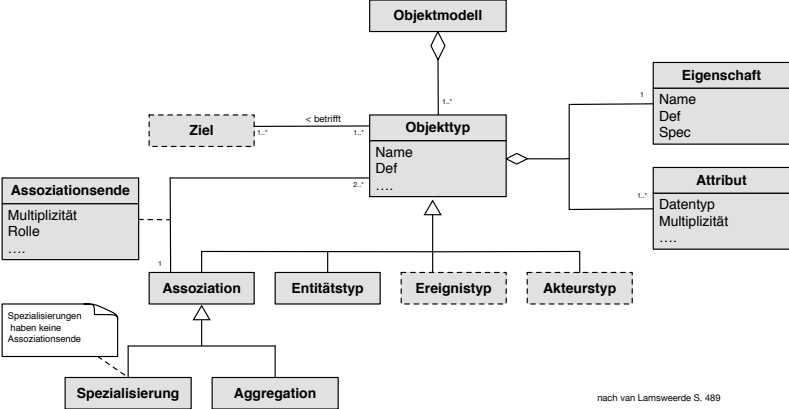
nach van Lamsweerde S. 487

# Das Zielemodell



nach van Lamsweerde S. 488

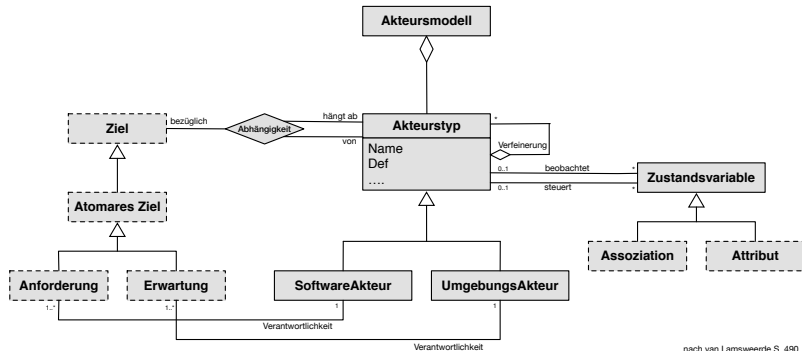
# Das Objektmodell



nach van Lamsweerde S. 489

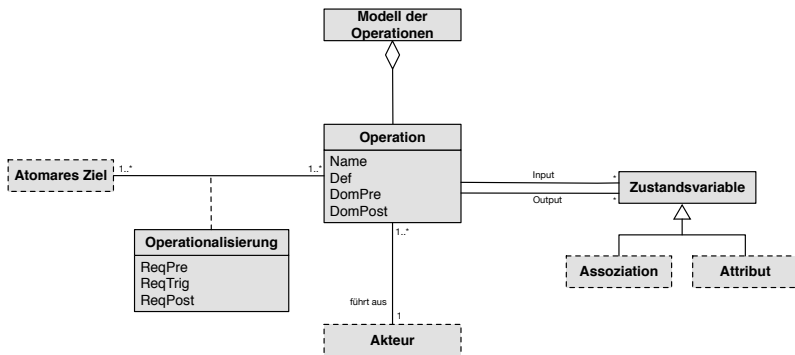


# Das Akteursmodell



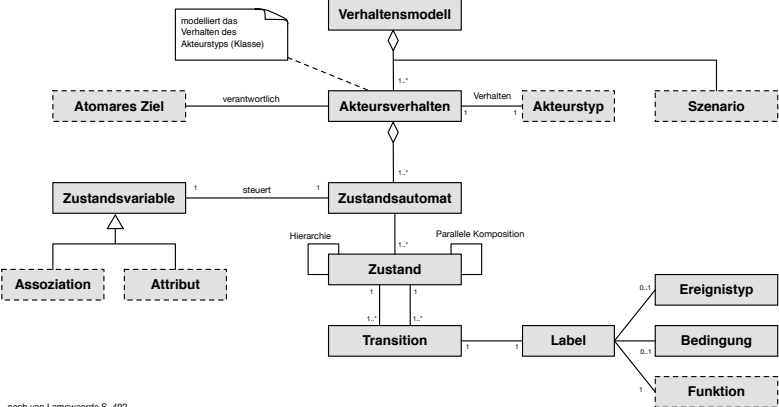
nach van Lamsweerde S. 490

# Das Modell der Operationen



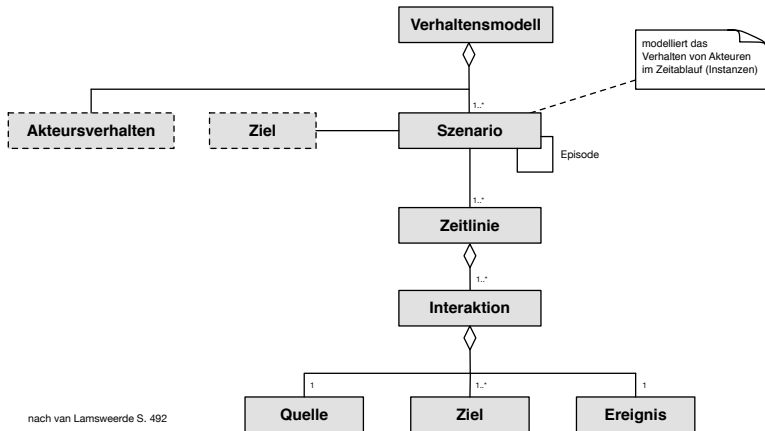
nach van Lamsweerde S. 491

# Das Verhaltensmodell, 1



nach van Lamsweerde S. 492

# Das Verhaltensmodell, 2



nach van Lamsweerde S. 492

# Konsistenzbedingungen für die Modelle

Die UML-Diagramme veranschaulichen nur einen Teil der Struktur und der Eigenschaften des Systemmodells, es gehören auch sehr viele **Konsistenzbedingungen** dazu.

Beispiele:

- Jedes konzeptuelle Objekt in der Beschreibung eines Ziel muss als Attribut oder Objekt im Objektmodell erscheinen
- Verhaltensziele können nur durch Verhaltensziele verfeinert werden
- Jeder Akteurstyp muss für mindestens ein atomares Ziel verantwortlich sein
- und viele mehr

# Übersicht

- Vorgehen
- Struktur des Modells – Metamodell
- Generierung des Anforderungsdokuments

# Generierung des Anforderungsdokument

- Eines der Ziele des Requirements Engineering ist die Erstellung eines hochwertigen **Anforderungsdokument**s
- Auf Basis eines zieleorientierten Modells kann es **halbautomatisch** erstellt werden
- Die **Definitionen** im Modell beschreiben große Teile des benötigten Dokuments
- Weitere Informationen müssen ergänzt werden
- Im Folgenden eine denkbare **Gliederung** und ihr Zusammenhang zum Modell:

# Gliederung I

## 1. Einführung

### 1.1 Zweck des Dokuments

### 1.2 Zweck des Systems

*Oberste Ziele mit ihren Defs aus dem Modell übernehmen und durch erläuternden Text ergänzen.*

### 1.3 Definitionen, Akronyme und Abkürzungen

*Das Glossar kann aus dem Objektmodell generiert werden. Der Abschnitt kann durch Ausschnitte von Diagrammen aus dem Objektmodell angereichert werden.*

### 1.4 Referenzen

*Liste der verwendeten und zitierten Dokumente, muss manuell erstellt werden.*



# Gliederung II

## 1.5 Überblick

*Darstellung der Organisation des Anforderungsdokuments.*

## 2. Allgemeine Beschreibung

### 2.1 Systemumgebung

*Basis ist das Akteursmodell, das die Umgebungsakteure und Softwareakteure enthält sowie die Schnittstellen zwischen den Akteuren. Der Text kann durch Abbildungen aus dem Akteursmodell ergänzt werden.*

### 2.2 Ziele

*Basis ist das Zielemodell, das in seinen (Teil-)Hierarchien top-down dargestellt und durch seine Defs erläutert wird. Zusätzlich zeigen Abbildungen die Verfeinerung der Ziele. Hier stehen die Oberziele und die weichen Ziele im Vordergrund.*

## Gliederung III

### 2.3 Charakteristik der Benutzer

*Text zu Annahmen über die Anwender, wie Erfahrung, technische Expertise etc.*

### 2.4 Allgemeine Randbedingungen

*Text über Randbedingungen für Entwicklung und Betriebs des Systems, die Auswirkungen auf die Qualität haben (können).*

### 2.5 Annahmen und Abhängigkeiten

*Aus dem Zielemodell und der Analyse der Hindernisse ergeben sich die Annahmen und Erwartungen, die für das System notwendig sind. Sie können aus dem Zielemodell übernommen werden. Auch ungelöste Hindernisse sollten hier aufgeschrieben werden.*

# Gliederung IV

## 2.6 Hinweise zur Weiterentwicklung

*Ziele, die mit einer niedrigen Priorität versehen wurden und in der nächsten Version des Systems (noch) nicht implementiert werden, sollten hier mit ihren Defs aufgezeichnet werden.*

## 3. Softwareanforderungen

### 3.1 Ziele und Anforderungen

*Das Zielemodell wird ausgehend von Ober- bzw. Zwischenzielen erläutert und bis zu den Akteuren dargestellt.*

*Die Aufteilung wird dem jeweiligen Projekt entsprechend gemacht.*

*Konflikte, Hindernisse und ihre Überwindung werden auch dargestellt.*

# Gliederung V

## 3.2 Objektmodell

*Die Struktur des Anwendungsgebiets sowie die dazu korrespondierenden Objekte im zu konstruierenden System werden dargestellt.*

*Dazu eignen sich Diagramme, die Ausschnitte des Objektmodells zeigen.*

## 3.3 Verantwortlichkeiten

*Alle Softwareakteure werden dargestellt und es wird beschrieben, für welche Anforderungen sie verantwortlich sind; dies ergibt sich aus dem Akteursmodell.*

*Ferner werden die Akteure der Umgebung aufgeführt, deren Agieren für das Erreichen der Ziele ebenso erforderlich ist.*

# Gliederung VI

## 3.4 Operationen

*Basis dieses Abschnitts ist das Modell der Operationen, deren Spezifikation pro ausführendem Akteur dargestellt werden.*

*Der Abschnitt kann durch ein Anwendungsfalldiagramm ergänzt werden.*

## 3.5 Verhalten

*Basis dieses Abschnitts ist das Verhaltensmodell. Zu den Zustandsdiagrammen werden beispielhafte Szenarien dargestellt.*

## 3.6 Zusammenfassung

*Eine Liste der Anforderungen mit ihren Definitionen.*

# Gliederung VII

## 3.6.1 Funktionale Anforderungen

*Die Liste der funktionalen Anforderungen kann aus dem Zielemodell hergeleitet werden, es handelt sich um alle Anforderungen der Kategorie ‚funktional‘.*

## 3.6.2 Nicht-funktionale Anforderungen

*Die Liste der nicht-funktionalen Anforderungen kann analog zu den funktionalen Anforderungen aus dem Zielemodell ermittelt werden.*

## Anhang

*In den Anhang kommt eine Referenz auf das Modell selbst.*