

Softwareanforderungsanalyse

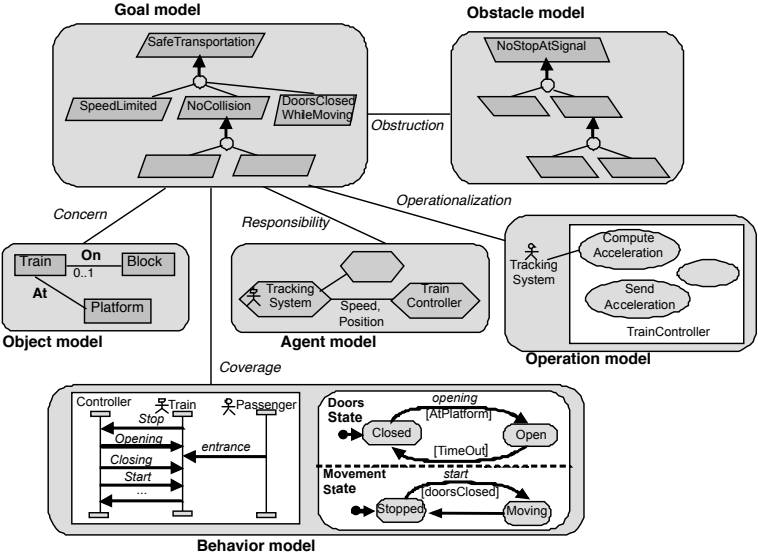
Modellierung der Operationen des Systems

Burkhardt Renz

THM, Fachbereich MNI

Wintersemester 2018/19

Das Modell der Operationen im Kontext der Modellierung



Übersicht

- Aufgaben des Modells der Operationen
- Sicht auf die Funktionalität
- Operationalisierung von Zielen
- Darstellung von Operationen im Modell
- Heuristische Regeln

Modellierung der Operationen des Systems

- Sicht auf die **Funktionalität**
 - Welche **Dienste** stellt das System bereit?
 - Wie tragen sie zum Erreichen von Zielen bei?
 - Die **Was**-Dimension der Anforderungsanalyse
- Funktionale Sicht wird benötigt für
 - Beschreibung von Leistungen für Anwender, externe Systeme etc.
 - Verfolgbarkeit der Realisierung von Zielen in den Operationen des Systems
 - Input für Design und Entwicklung
 - Basis für Definition von Testfällen, insbesondere für Black-Box-Tests
 - Basis für die Dokumentation für Benutzer
- Darstellung: Operationalisierungsdiagramm, das zeigt, wie **Operationen** zum Erreichen von **Zielen** beitragen und **Objekte** im Objektmodell dazu als Input/Output verwenden

Übersicht

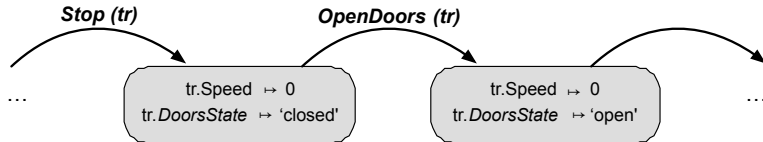
- Aufgaben des Modells der Operationen
- **Sicht auf die Funktionalität**
 - Was sind Operationen?
 - Charakterisierung von Operationen
- Operationalisierung von Zielen
- Darstellung von Operationen im Modell
- Heuristische Regeln

Was sind Operationen?

- Operationen verändern den Zustand des Systems
- Ihre Signatur besteht aus den Eingabevariablen und den Ausgabevariablen
- Man kann sie also auffassen als Menge von Paaren von Zuständen *vorher* – *nachher*

$$Op \subseteq InputState \times OutputState$$

- Die Ausführung der Operation entspricht dann genau einem Übergang von einem *InputState* in einen *OutputState*



Eigenschaften von Operationen

Eigenschaften

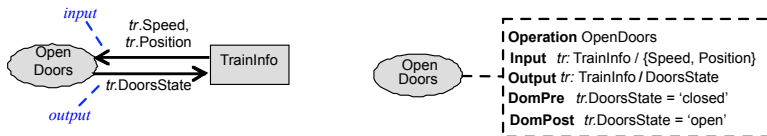
- Operationen müssen Ziele des Zielemodells **operationalisieren**
- Operationen sind im Modell **atomar**
- Akteure agieren nebenläufig und auch ein einzelner Akteur kann mehrere Operationen **parallel** ausführen

Kategorien von Operationen

- **Softwareoperationen** ausgeführt durch Softwarekomponenten des Systems, oft auch Dienste oder **Services** genannt
- Operationen der **Umgebung** des Systems, ausgeführt durch menschliche Akteure, Geräte oder externe Softwaresysteme, oft auch **Tasks** genannt

Charakterisierung von Operationen

- Annotierte Grundeigenschaften: *Name*, *Def*, *Category*
- **Signatur**, d.h. Input und Output textuell oder graphisch
- **Vor- und Nachbedingungen** in Bezug auf das Anwendungsgebiet beschreiben (*deskriptiv*) den Zustand des Systems vor und nach der Ausführung der Operation.
- Welcher **Akteur** führt die Operation aus?



Quelle: Lamsweerde S. 425, 426

Ausführender der Operation

- Ein **Akteur** führt eine Operation aus, wenn Instanzen des Akteurs die Ausführung anstoßen können. Dazu muss gelten:
- Ein ausführender Akteur muss die Eingabe- und Ausgabevariablen der Operation beobachten bzw. steuern.
- Eine Operation wird von genau einem Akteur (dessen Instanzen) ausgeführt.

Übersicht

- Aufgaben des Modells der Operationen
- Sicht auf die Funktionalität
- **Operationalisierung von Zielen**
 - Vor-, Nach- und Triggerbedingungen für die Zielerreichung
 - Verhalten von Akteuren
 - Operationalisierung von Zielen und Erfüllungsargument
- Darstellung von Operationen im Modell
- Heuristische Regeln

Operationalisierung von Zielen

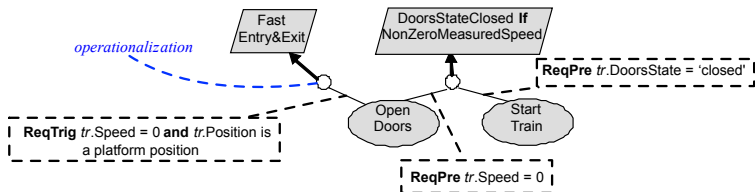
- **Ziele** wurden im Zielemodell so weit verfeinert, dass für ihre Erreichung genau **ein Akteur** verantwortlich ist
- **Akteure** können **Operationen** ausführen.
- Man spricht von der **Operationalisierung eines Ziels** um zu bezeichnen, dass die Operationen, die der für das Ziel verantwortliche Akteur ausführen kann, dazu führen, dass das Ziel erreicht wird.
- graphisch wird das dargestellt durch eine Verbindung zwischen dem Ziel und der Operation.

Präskriptive Bedingungen für Zielerreichung

Erforderliche Bedingungen (*required conditions*) sollen sicherstellen, dass das gewünschte Ziel erreicht wird

- ReqPre: Vorbedingung, die die Eingabevariablen erfüllen müssen, damit die Operation ausgeführt werden *kann*
- ReqTrig: Triggerbedingung bedeutet, dass die Operation ausgeführt werden *muss*, wenn die Bedingung eintritt
- ReqPost: gibt an, was nach der Ausführung der Operation garantiert ist

Beispiel für Operationalisierung eines Ziels



Quelle: Lamsweerde S. 428

Das Beispiel textuell

Operation OpenDoors

```
Def Software operation controlling the opening of
  all doors of a train;
Input tr: TrainInfo;
Output tr: TrainInfo/DoorsState;
DomPre tr.DoorsState = 'closed';
DomPost tr.DoorsState = 'open';
ReqPre for DoorsStateClosedIfNonZeroMeasuredSpeed:
  tr.Speed = 0;
ReqPre for SafeEntry&Exit:
  tr.Position is a platform position;
ReqTrig for FastEntry&Exit:
  tr.Position is a platform position and tr.Speed = 0;
```

Verhalten von Akteuren

- Zur Erreichung der Ziele, für die ein Agent verantwortlich ist, muss er die Operationen ausführen, die das Ziel operationalisieren und zwar entsprechend der Vor-, Nach- und Triggerbedingungen.
- Ein Akteur, der eine Operation sofort ausführt, wenn alle ihre Vorbedingungen erfüllt sind, wird **eifrig** (*eager*) genannt.
- Ein Akteur, der eine Operation erst ausführt, wenn eine Triggerbedingung das verlangt, wird **verzögert** (*lazy*) genannt.
- Mehrere Triggerbedingungen können im selben Systemzustand zutreffen – die dadurch ausgelösten Operationen werden dann *nebenläufig* ausgeführt.

Operationalisierung von Zielen und Erfüllungsargument

Ein Ziel G ist korrekt **operationalisiert** in die Operationen Op_1, \dots, Op_n wenn die Spezifikationen der Operationen $Spec(Op_1), \dots, Spec(Op_n)$ folgendes erfüllen:

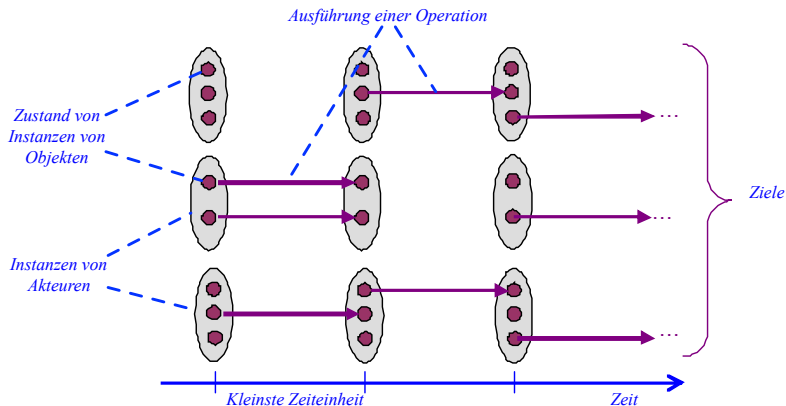
- 1 Vollständigkeit, d.h.
 $\{Spec(Op_1), \dots, Spec(Op_n)\} \models G$
- 2 Konsistenz d.h.
 $\{Spec(Op_1), \dots, Spec(Op_n)\} \not\models \perp$
- 3 Minimalität, d.h.
Für jede echte Teilmenge $S \subset \{Spec(Op_1), \dots, Spec(Op_n)\}$
gilt $S \not\models G$

Das **Erfüllungsargument** besteht im Nachweis, dass $\{Spec(Op_1), \dots, Spec(Op_n)\} \models G$ gilt.

Übersicht

- Aufgaben des Modells der Operationen
- Sicht auf die Funktionalität
- Operationalisierung von Zielen
- **Darstellung von Operationen im Modell**
 - Ziele, Akteure, Objekte und Operationen
 - Operationalisierungsdiagramm
 - Das Anwendungsfalldiagramm der UML
- Heuristische Regeln

Wie hängen Ziele, Akteure, Objekte und Operationen zusammen?



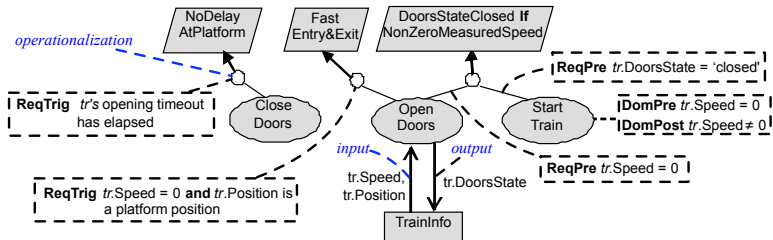
Quelle: Lamsweerde S.434

Operationalisierungsdiagramm

Das Operationalisierungsdiagramm zeigt in einem annotierten Graph, durch welche Operationen Ziele erreicht werden:

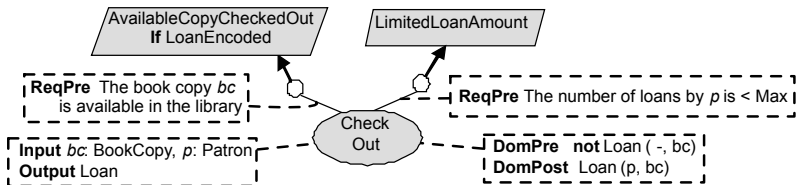
- Operationen, Ziele und Objekte sind die Knoten
- Operationalisierung verbindet Operationen und Ziele
- Input/Output-Beziehungen verbinden Operationen mit Objekten

Operationalisierungsdiagramm – Beispiel Zugsteuerung



Quelle: Lamsweerde S.435

Operationalisierungsdiagramm – Beispiel Bibliothek



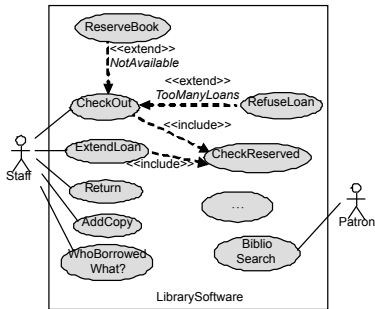
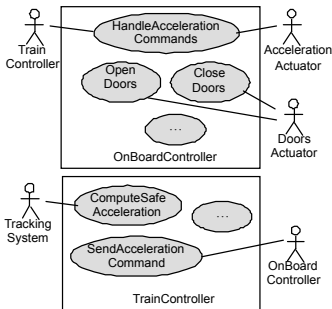
Quelle: Lamsweerde S.436

Das Anwendungsfalldiagramm der UML

Das Anwendungsfalldiagramm der UML kann aus den Operationalisierungsdiagrammen generiert werden:

- Für jeden Akteur sind alle Operationen, die er ausführen kann seine Anwendungsfälle.
- Andere Akteure sind beteiligt, wenn sie Attribute von Objekten beobachten oder steuern, die in der Ausführung der Operationen vorkommen.

Operationen und UML Anwendungsfalldiagramm – Beispiel



Quelle: Lamswerde S.436

Übersicht

- Aufgaben des Modells der Operationen
- Sicht auf die Funktionalität
- Operationalisierung von Zielen
- Darstellung von Operationen im Modell
- Heuristische Regeln

Vorgehen beim Entwickeln der Operationalisierungsdiagramme

- 1 Operationen identifizieren mit DomPre und DomPost
- 2 Identifizieren und Zuordnen der Input- und Output-Variablen
- 3 Präzise Spezifikation der Operationen mit Vor-, Nach- und Triggerbedingungen
- 4 Operationalisierungsbeziehungen zu den Zielen im Zielemodell herstellen

Heuristiken

- Analyse von Zielen:
 - Ist das Ziel vom Typ `Achieve`
 - Operation, die den gewünschten Zustand erzeugt.
 - Ist das Ziel vom Typ `Maintain`
 - Operation, die die Bedingungen sicherstellt.
- Szenarien durchspielen und daraus Operationen entwickeln.
- Zusammenhang zwischen Zielen, Objekten und Akteuren untersuchen, um die verbindenden Operationen zu finden.