

Objektorientierte Analyse

Vorlesung Mikrorechentchnik 2

Sommersemester 2020

Agenda



Objektorientierte Analyse und Design (OOA/OOD) an dem Systembeispiel **Verteilte Ampelsteuerung**

- Wiederholung: Merkmale der Objektorientierung
- Neu: Systembeispiel verteilte Ampelsteuerung
- Neu: Methodisches Vorgehen mit Unified Modeling Language (UML)

Lernziele

- Sie können die wesentlichen **Prinzipien der Objektorientierung benennen** und **beschreiben**
 - Objekt-Klasse, Kapselung, Objektidentität, Kohärenz, Vererbung, Zusicherung-Verantwortlichkeit
- Sie können die wesentlichen Schritte der vorgestellten Methodik zur objektorientierten Analyse und Design mit Unified Modeling Language **wiedergeben** und für einfache Sachverhalte **anwenden**
 - Analyse: Systemidee & Zielsetzung, Systemkontext, Anwendungsfall, Fachklassen, Systemablaufmodell
 - Design: Design, Architektur, Schichtenmodell, Verteilungsmodell, Subsystemmodell, Klassenmodell, Zustandsmodell
- Sie können die wesentlichen Elemente der verteilten Ampelsteuerung mit UML **skizzieren**

Objektorientierte Analyse – diese Vorlesung (nach Oestereich 2012)



- Schritt 1: Systemidee und Zielsetzung entwickeln
- Schritt 2: Interessenhalter ermitteln
- Schritt 3: Interessen der Interessenhalter identifizieren
- Schritt 4: Systemkontext aufnehmen
- Schritt 5: Geschäftsanwendungsfälle identifizieren
- Schritt 6: Anwendungsfälle essentiell beschreiben
- Schritt 7: Materialsammlung und -studie
- Schritt 8: Systemanwendungsfälle identifizieren
- Schritt 9: Fachklassen, Aufzählungen und Konfigurationswerte identifizieren
- Schritt 10: Fachliches Glossar anlegen
- Schritt 11: Systemablaufmodelle entwickeln
- Schritt 12: Systemanwendungsfallmodell erstellen
- Schritt 13: Übrige fachliche Anforderungen und Regeln beschreiben
- Schritt 14: Systemschnittstelle beschreiben
- Schritt 15: Exploratives Schnittstellen-Prototyping

Viele der Schritte werden je nach Projekt parallel und iterativ durchgeführt .

Einige Schritte können ausgelassen werden

Objektorientiertes Design – nächste Vorlesung (nach Oestereich 2012)



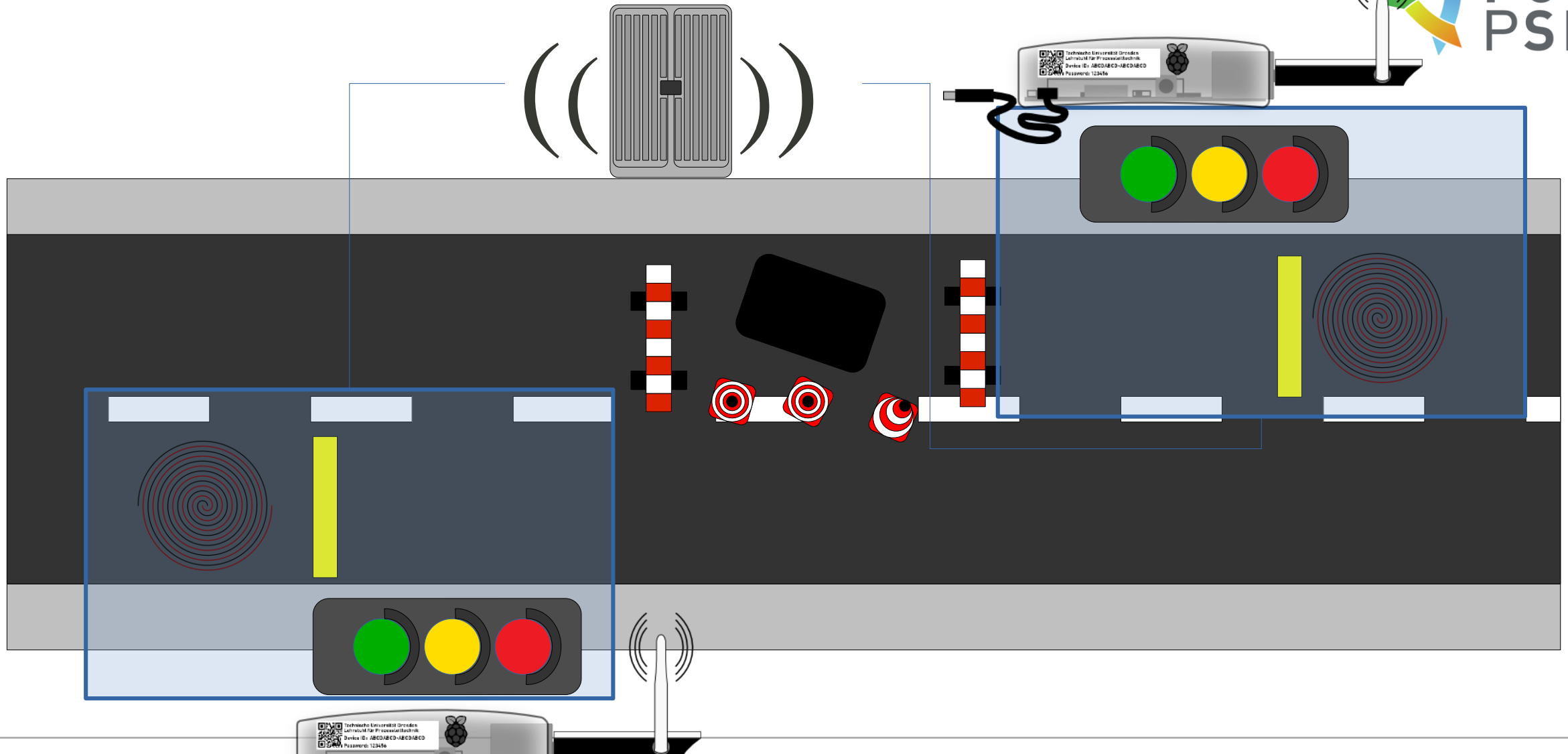
- Schritt 1: Schichtenmodell definieren
- Schritt 2: Verteilungsmodell definieren
- Schritt 3: Fachliches Subsystemmodell definieren
- Schritt 4: Ablaufverantwortlichkeit festlegen
- Schritt 5: Komponentenspezifische Klassenmodelle entwickeln
- Schritt 6: Komponentenschnittstellen entwerfen
- Schritt 7: Zustandsmodelle (weiter-)entwickeln
- Schritt 8: Objektfluss modellieren
- Schritt 9: Interaktionsmodelle entwickeln
- Schritt 10: Attribute definieren
- Schritt 11: Dialoge spezifizieren
- Schritt 12: Design-Diskurs
- Schritt 13: Testgetriebene Entwicklung
- Schritt 14: Ablauforientierte Komponententests entwickeln
- Schritt 16: Klassentests entwickeln

Viele der Schritte werden je nach Projekt parallel und iterativ durchgeführt .

Wir werden ausgewählte Schritte für unser Fallbeispiel durchführen

Objektorientierte Analyse Verteilte Ampelsteuerung mit Industrie-4.0-Elementen

Grundidee

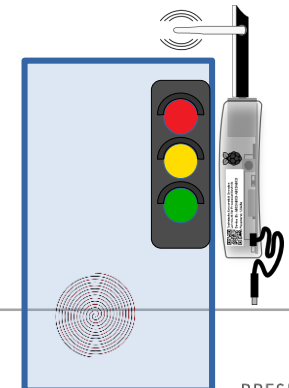
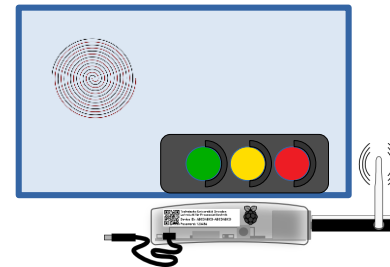
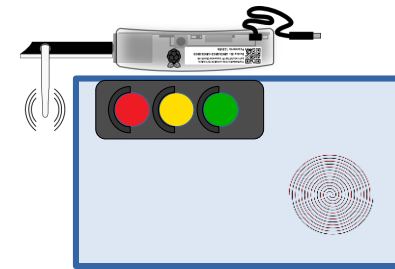
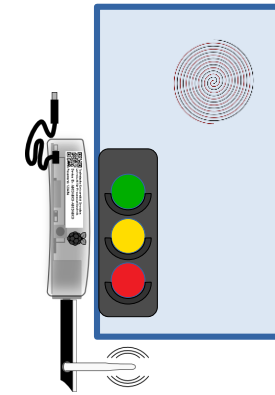


Schritt 1: Systemidee und Zielsetzung entwickeln

Aus einer vagen Idee

Eine Ampelanlage soll schnell und einfach aus vernetzten intelligenten Ampelmodulen aufgebaut werden können

eine konkrete Systemidee und Zielsetzung ableiten



Schritt 1: Systemidee und Zielsetzung entwickeln

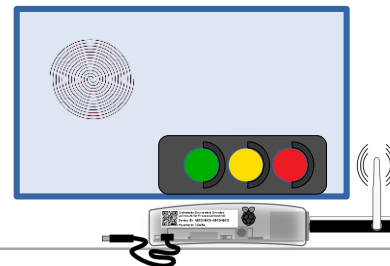
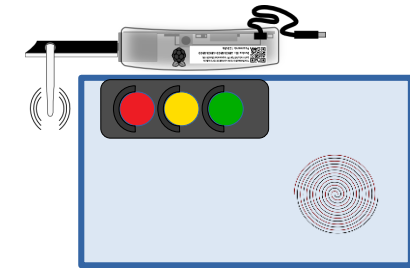
Erstellen eines Produktkartons

Produktkartons enthalten üblicherweise folgende Information

- Systemvoraussetzungen
- Wichtige Eigenschaften
- Was leistet die Software für den Nutzer?

Aufgabe: Produktkarton erstellen um die vage Idee zu präzisieren

- Eine Ampelanlage soll schnell und einfach aus intelligenten Ampelmodulen aufgebaut werden können



Schritt 1: Systemidee und Zielsetzung entwickeln

Erstellen eines Produktkartons - Ampelbeispiel



Das Ampelprogramm läuft auf jedem eingebetteten Systemen mit folgenden Eigenschaften

- $\mu\text{C} > 400 \text{ MHz}$
- $\text{RAM} > 10 \text{ MByte}$
- TCP/IP-Stack und WLAN-Schnittstelle
- Peripherie zur Ansteuerung von drei Lichtern und Aufnahme von einem Eingangssignal

Portabel, Kosteneffizient, Erweiterbar

Einfachste Orchestrierung

- Schritt 1: Ampelanlage aus Ampelmodulen physisch aufbauen
- Schritt 2: Verhalten der Ampelanlage konfigurieren
- Schritt 3: Ampelanlage testen und in Betrieb nehmen

Achtung: Spiel- und Lernsystem, nicht für ernsthafte Anwendungen geeignet

Schritt 1: Systemidee und Zielsetzung entwickeln

Erstellen eines Produktkartons – Leitfragen, Checkliste



- Umfang ~ ½ Seite?
- Beschreibt die Idee, was mit dem zu entwickelnden System erreicht werden soll?
- Ist die Systemidee so formuliert, dass die Fertigstellung des Produktes geistig vorweggenommen wurde?
- Enthält die Systemidee eine Auflistung der 5-15 wichtigsten Eigenschaften und Leistungsmerkmale?
- Enthält die Systemidee eine Auflistung der wichtigsten Voraussetzungen und Rahmenbedingungen?
- Enthält die Systemidee eine Abgrenzung, welche Eigenschaften erwartet werden können, jedoch explizit nicht oder erst in einer späteren Version enthalten sein werden?

Schritt 1: Systemidee und Zielsetzung entwickeln

Prüfen des Produktkarton mit der Checkliste



- Umfang ~ ½ Seite?
- Beschreibt die Idee, was mit dem zu entwickelnden System erreicht werden soll?
- Ist die Systemidee so formuliert, dass die Fertigstellung des Produktes geistig vorweggenommen wurde?
- Enthält die Systemidee eine Auflistung der 5-15 wichtigsten Eigenschaften und Leistungsmerkmale?
- Enthält die Systemidee eine Auflistung der wichtigsten Voraussetzungen und Rahmenbedingungen?
- Enthält die Systemidee eine Abgrenzung, welche Eigenschaften erwartet werden können, jedoch explizit nicht oder erst in einer späteren Version enthalten sein werden?

Das Ampelprogramm läuft auf jedem eingebetteten Systemen mit folgenden Eigenschaften

- $\mu\text{C} > 400 \text{ MHz}$
- $\text{RAM} > 10 \text{ MByte}$
- TCP/IP-Stack und WLAN-Schnittstelle
- Peripherie zur Ansteuerung von drei Lichtern und Aufnahme von einem Eingangssignal

Portabel, Kosteneffizient, Erweiterbar

Einfachste Orchestrierung

- Schritt 1: Ampelanlage aus Ampelmodulen physisch aufbauen
- Schritt 2: Verhalten der Ampelanlage konfigurieren
- Schritt 3: Ampelanlage testen und in Betrieb nehmen

Achtung: Spiel- und Lernsystem, nicht für ernsthafte Anwendungen geeignet

Schritt 2: Interessenthalter ermitteln



2-1: Identifiziere die Interessenthalter

2-2: Bewerte Wichtigkeit anhand von Relevanz und Risiko

2-3: Identifiziere die Projektansprechpartner

2-4: Unterscheide zwischen Fachexperten, Anforderungsverantwortlichen und Systembetroffenen

Schritt 2: Interessenthalten identifizieren

2-1: Mögliche Interessenthalten



Interessenthalten: alle Parteien, die von dem System/Projekt direkt oder indirekt betroffen sind oder von denen das System/Projekt direkt oder indirekt betroffen ist

- Anwender des Systems
- Fachabteilung
- Revisionsabteilung
- Auftraggeber, Geldgeber, Management, Vorstand, Geschäftsführung
- Gesetzgeber, Standards
- Kunden
- Systemadministratoren, Servicepersonal, Schulungspersonal, Hotline, Support
- Systementwickler, Systemwartung
- Käufer
- Marketing, Vertrieb
- Projektgegner und -befürworter

Schritt 2: Interessenhalter identifizieren

2-2: Interessenhalter bewerten



In der Praxis nur begrenzte Ressourcen für Kommunikation mit Interessenhaltern

- Wie groß ist das Risiko, wenn dieser Interessenhalter nicht berücksichtigt wird?
- Wie groß ist der Aufwand, diesen Interessenhalter zu berücksichtigen?

Beispiel Ampelsystem für MRT-2

Risiko \ Aufwand	Gering	Mittel	Hoch
Hoch			
Mittel			
Gering			

Schritt 2: Interessenthaler identifizieren

2-3: Identifiziere Projekt-Ansprechpartner



Fachexperten

- Experten des Anwendungsgebiets
- Können fachliche Fragen qualifiziert beantworten

Anforderungsverantwortliche

- Befugt Anforderungen festzulegen bzw. abzustimmen
- (zumeist keine Fachexperten oder Systembetroffene)

Systembetroffene

- Spätere Anwender

Schritt 3: Interessen der Interessenthalten identifizieren



3-1: Beschreibe die Ziele und Interessen der einzelnen Interessenthalten

3-2: Identifiziere bestehende Probleme und Schwachstellen aus Sicht der Interessenthalten

3-3: Beschreibe die wichtigen geforderten Systemeigenschaften aus Sicht der Interessenthalten

Checkliste

- Sind Interessen und Ziele aller wichtigen Interessenthalten ermittelt und dokumentiert?
- Für welches Fachgebiet ist Interessenthalten kompetent? Wie lange schon?
- Welche Befugnisse und Verantwortung trägt der Interessenthalten?
- Welches Interesse hat er am neuen System?
- Was muss das System aus seiner Sicht auf jeden Fall leisten?
- Was darf keinesfalls passieren?
- Welche Befürchtungen hat der Interessenthalten ggü. dem neuen System?

Schritt 4: Systemkontext aufnehmen



4-1: System darstellen (UML Systemkontextdiagramm)

4-2: Fremdsysteme ermitteln

4-3: Stelle alle Benutzer des Systems als Akteure dar

4-4: Modelliere die Beziehungen zwischen den Akteuren / Fremdsystemen und dem zu erstellenden System

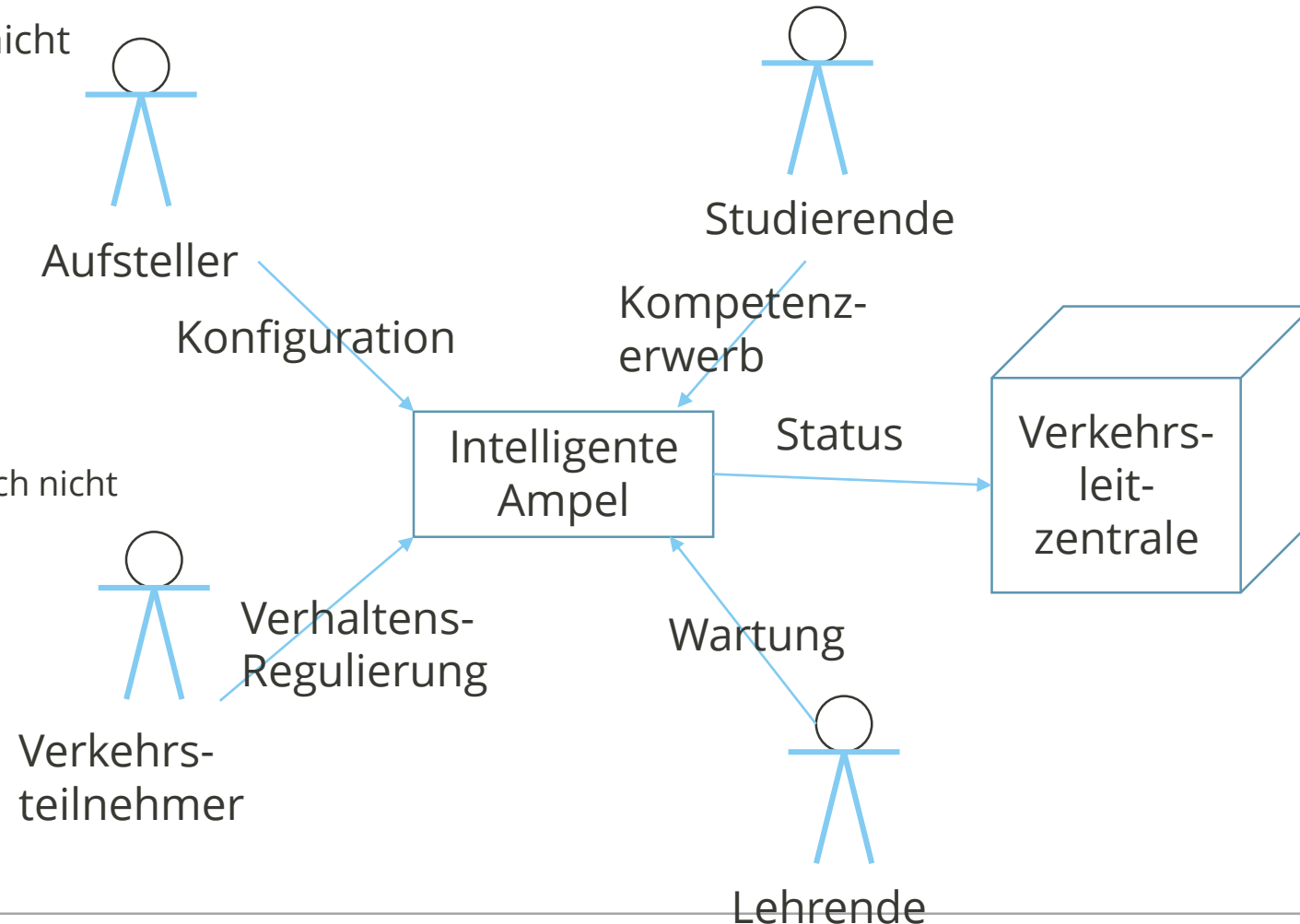
Schritt 4: Systemkontext aufnehmen

4-1 System darstellen bis 4-4 Modellieren

Ziel: klären, was zum System gehört und was nicht

Systemkontextdiagramm

- Menschliche Akteure
 - Ansprechpartner für uns
- Technische Systeme
 - haben Schnittstelle, deren Funktion gehört jedoch nicht zu unserem Leistungsumfang



Schritt 5: Geschäftsanwendungsfälle identifizieren



5-1: Entscheide, ob Geschäftsanwendungen identifiziert werden sollen

5-2: Identifiziere die **Geschäftsanwendungsfälle**

- Geschäftsanwendungsfall = Geschäftlicher Ablauf. Wird von einem geschäftlichen Ereignis ausgelöst und endet mit einem Ergebnis, das direkt oder indirekt einen **geschäftlichen Wert** darstellt
- Unabhängig von den konkreten Möglichkeiten und Anforderungen zu ihrer systemtechnischen Umsetzung

5-3: Identifiziere Anfang und Ende der Anwendungsfälle mit Hilfe von **Auslösern** und **Ergebnissen**

- Akteure, Auslöser und Ergebnis müssen konkret benannt werden

5-4: Identifiziere die auszuschließenden Geschäftsanwendungsfälle

- Gerne mitbeschreiben und explizit als *{excluded}* bezeichnen

Schritt 6: Anwendungsfälle essentiell beschreiben



- 6-1: Beschreibe die Essenz der hochprioren Geschäftsanwendungen
- 6-2: Unterscheide stabile und veränderliche Anforderungen
- 6-3: Definiere für jeden Anwendungsfall Auslöser, **Vorbedingungen** und **eingehende Information**
- 6-4: Definiere für jeden Anwendungsfall Ergebnisse, **Nachbedingungen** und **ausgehende Information**
- 6-5: Beschreibe mit jeden Anwendungsfall einen kohärenten fachlichen Sachverhalt
 - Zerlege ggf. in mehrere eigenständige Anwendungsfälle

Schritt 6: Essentielle Anwendungsfallbeschreibung - Ampel konfigurieren



Name	Einzelampel konfigurieren
Art	Geschäftsanwendungsfall
Kurzbeschreibung	Aufsteller konfiguriert Einzelampel
Auslöser	Auftrag zum Aufbau eines neuen Ampelsystems
Ergebnis	Konfigurierte Einzelampel
Akteure	Aufsteller
Eingehende Information	Ort und Funktion der Einzelampel
Vorbedingung	Ampelsystem ist vorgeplant
Nachbedingungen	Die Ampel ist konfiguriert und die Konfiguration in dem Ampel-Repository gesichert
Essenzielle Schritte	<ul style="list-style-type: none">• IP-Adresse einstellen• Ort und Funktion festlegen• Konfiguration sichern

Schritt 7: Materialsammlung und -studie

7-1: Identifiziere und studiere die Materialien, Gegenstände, Beispiele und Muster im Anwendungsbereich

- Formulare, Vordrucke, Arbeitsplatzbeschreibungen, Normen
- Datenblätter, elektrische Schaltungen, ... (Vorlesung MRT-1 ☺)

7-2: Bewerte die Materialien und Gegenstände hinsichtlich ihrer Brauchbarkeit und Relevanz für das aktuelle Projekt

Bewertung	?	++	+	0	-	--
Aktualität						
Verbindlichkeit						
Korrektheit						
Wichtigkeit						

Schritt 8: Systemanwendungsfälle identifizieren



8-1: Identifiziere die konkreten Systemanwendungsfälle. Falls Geschäftsanwendungsfälle identifiziert wurden: Entscheide, welche umgesetzt werden sollen

8-2: Zerlege die systemtechnisch umzusetzenden Geschäftsanwendungsfälle in zeitlich kohärente Systemanwendungsfälle

8-3: Ergänze ggf. weitere Information wie Ansprechpartner, Risiko, Wichtigkeit, Aufwand, Stabilität, etc.

Schritt 8: Systemanwendungsfallbeschreibung

Systemanwendungsfall				
Name	Einzelampel konfigurieren			
Kurzbeschreibung	Aufsteller konfiguriert Einzelampel			
Akteure	Aufsteller			
Auslöser	Aufbau eines neuen Ampelsystems			
Vorbedingung	Geplantes Gesamtampelsystem mit definierten Funktionen			
Eingehende Information	ID der Einzelampel, Funktion der Einzelampel			
Ergebnisse	Konfigurierte Einzelampel, Gesicherte Konfiguration			
Nachbedingungen	Die Einzelampel ist bereit zur Integration in das Ampelsystem			
Ablauf	<ol style="list-style-type: none"> 1) Einzelampel mit Rechner verbinden <ul style="list-style-type: none"> - über Ethernet-Kabel anschließen, 192.168.0.240 2) Einzelampel identifizieren <ul style="list-style-type: none"> - mac-adresse auslesen 3) Konfiguration einstellen <ul style="list-style-type: none"> - hardwareswitches oder Softwareparameter einstellen 4) Konfiguration auslesen, prüfen und abspeichern 			
Ansprechpartner	Herr Iatrou			
Risiko	noch nicht betrachtet			
Verbindlichkeit, Prio	hohe Prio, unverzichtbar			
Aufwand	noch nicht geschätzt			
Stabilität	instabil, Hardware und Ablauf unter Review			
Zeitpunkt, Dringlichkeit	Release 0.8			
Änderungshistorie				
Datum	Mitarb	Status	Kommentar	
09.04.2018	LU	Entwurf	Erster, nicht abgestimmter Entwurf	

Schritt 9: Fachklassen, Aufzählungen und Konfigurationswerte identifizieren



9-1: Identifiziere die wichtigsten fachlichen Gegenständen, die vom System repräsentiert werden sollen,

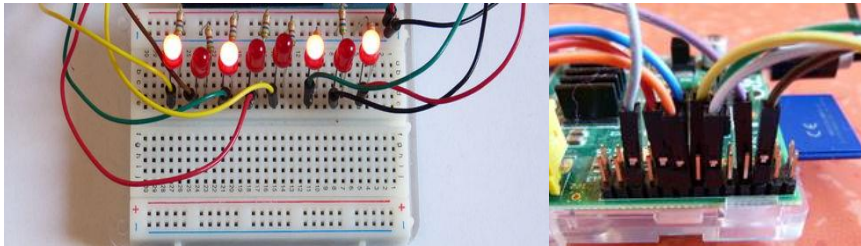
9-2: Betrachte diese Gegenstände als Klassen und modelliere ihre **strukturellen Zusammenhänge** in einem **Klassendiagramm**

9-3: Gib den Klassen sprechende Namen, benenne ihre **Assoziationen** und **Assoziationsrollen** und beschreibe soweit möglich ihre **Multiplizitäten**

9-4: Beschreibe geschäftliche Konstanten, Standardwerte und Aufzählungsmengen in Form von Klassen

Schritt 9-1: Identifiziere wichtigste fachliche Gegenstände

Fachliche Gegenstände der Einzellampe



Address	Field Name	Description	Size	Read/Write
0x 7E20 0000	GPFSEL0	GPIO Function Select 0	32	R/W
0x 7E20 0000	GPFSEL0	GPIO Function Select 0	32	R/W
0x 7E20 0004	GPFSEL1	GPIO Function Select 1	32	R/W
0x 7E20 0008	GPFSEL2	GPIO Function Select 2	32	R/W
0x 7E20 000C	GPFSEL3	GPIO Function Select 3	32	R/W
0x 7E20 0010	GPFSEL4	GPIO Function Select 4	32	R/W
0x 7E20 0014	GPFSEL5	GPIO Function Select 5	32	R/W
0x 7E20 0018	-	Reserved	-	-
0x 7E20 001C	GPSET0	GPIO Pin Output Set 0	32	W
0x 7E20 0020	GPSET1	GPIO Pin Output Set 1	32	W
0x 7E20 0024	-	Reserved	-	-
0x 7E20 0028	GPCLR0	GPIO Pin Output Clear 0	32	W
0x 7E20 002C	GPCLR1	GPIO Pin Output Clear 1	32	W
0x 7E20 0030	-	Reserved	-	-
0x 7E20 0034	GPLEV0	GPIO Pin Level 0	32	R
0x 7E20 0038	GPLEV1	GPIO Pin Level 1	32	R
0x 7E20 003C	-	Reserved	-	-
0x 7E20 0040	GPEDS0	GPIO Pin Event Detect Status 0	32	R/W
0x 7E20 0044	GPEDS1	GPIO Pin Event Detect Status 1	32	R/W

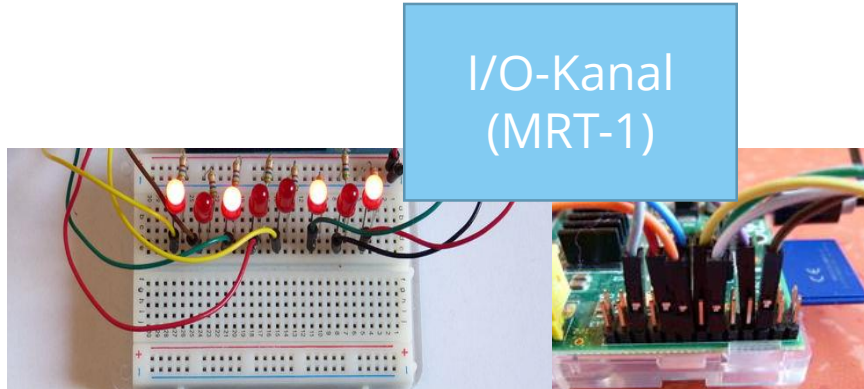
```
#include „bcm2835.h“
```

```
int main() {
    int status;
    if ( status = bcm2835_init() >= 0) { // initialisiere library
        bcm2835_gpio_definePin(15, BCM2835_GPIO_OUT); // definiere Funktion P
        bcm2835_gpio_setPin(15); // setze Pin auf 1
    }
    return status;
}
```



Schritt 9-2: Betrachte Gegenstände als Fachklassen

Abbildung der Gegenstände der Einzelampel auf Fachklassen

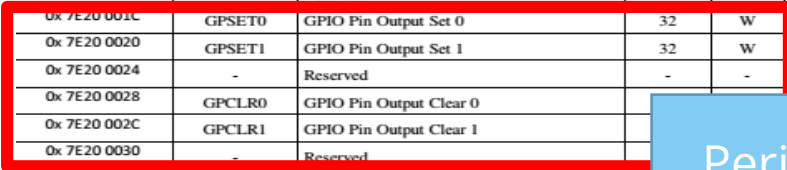


I/O-Kanal (MRT-1)

IoT-Server



Address	Field Name	Description	Size	Read/Write
0x 7E20 0000	GPFSEL0	GPIO Function Select 0	32	R/W
0x 7E20 0000	GPFSEL0	GPIO Function Select 0	32	R/W
0x 7E20 0004	GPFSEL1	GPIO Function Select 1	32	R/W
0x 7E20 0008	GPFSEL2	GPIO Function Select 2	32	R/W
0x 7E20 000C	GPFSEL3	GPIO Function Select 3	32	R/W
0x 7E20 0010	GPFSEL4	GPIO Function Select 4	32	R/W
0x 7E20 0014	GPFSEL5	GPIO Function Select 5	32	R/W
0x 7E20 0018	-	Reserved	-	-
0x 7E20 001C	GPSET0	GPIO Pin Output Set 0	32	W
0x 7E20 0020	GPSET1	GPIO Pin Output Set 1	32	W
0x 7E20 0024	-	Reserved	-	-
0x 7E20 0028	GPCLR0	GPIO Pin Output Clear 0	32	W
0x 7E20 002C	GPCLR1	GPIO Pin Output Clear 1	32	W
0x 7E20 0030	-	Reserved	-	-
0x 7E20 0034	GPLEV0	GPIO Pin Level 0	32	R/W
0x 7E20 0038	GPLEV1	GPIO Pin Level 1	32	R/W
0x 7E20 003C	-	Reserved	-	-
0x 7E20 0040	GPEDS0	GPIO Pin Event Detect Status 0	32	R/W
0x 7E20 0044	GPEDS1	GPIO Pin Event Detect Status 1	32	R/W



Peripherie (MRT-1)

```
#include „bcm2835.h“
```

```
int main() {
    int status;
    if ( status = bcm2835_init() >= 0) { // initialisiere library
        bcm2835_gpio_definePin(15, BCM2835_GPIO_OUT); // definiere Funktion P
        bcm2835_gpio_setPin(15); // setze Pin auf 1
    }
    return status;
}
```

Zustands-automat

Schritt 9-2: Benenne Klassen, Assoziationen, Multiplizitäten

Analyse-Fachklassenmodell der Ampelkomponente

Fachklassenmodell

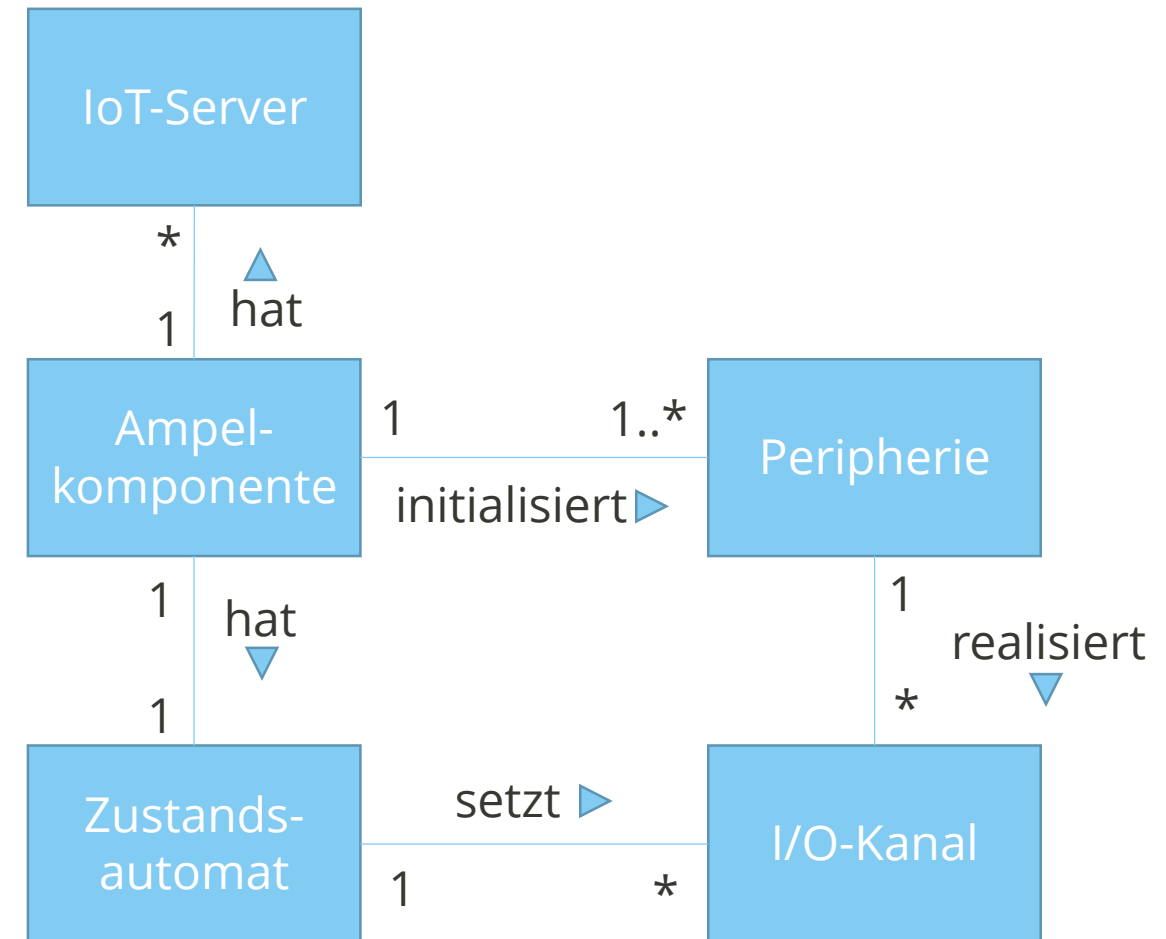
- Keine Details und Feinheiten
- Werden während Design häufig weiter zerlegt

Assoziationen mit Multiziplicitätsangaben

- Zulässige Mengenangaben, spiegelt Fachwissen der Interessenhalter wieder
- 1 genau 1
- 0..1 0 oder eins
- 0..4 0 bis vier
- * größer oder gleich 0 (default)
- 1..* größer oder gleich eins

Leserichtung durch Dreiecke

- Genau eine (=jede) Ampelkomponente hat genau einen Zustandsautomat
- Jede Ampelkomponente hat beliebig viele IoT-Server
- Jede Ampelkomponente initialisiert 1 oder mehr Peripherie
- Jeder Zustandsautomat setzt beliebig viele I/O-Kanäle
- Jede Peripherie realisiert beliebig viele I/O-Kanäle



Zusammenfassung



Prozessmodell für die Objektorientierte Analyse und Design

- Objektorientierte Analyse: Was will ich erreichen?
 - Ziele festlegen, Stakeholder benennen, Fachklassenkonzept entwickeln
- Objektorientiertes Design: Wie will ich das Ziel erreichen?
 - Konzepte auf Datenstrukturen und Algorithmen abbilden

Literaturempfehlungen

- Oestereich, Scheithauer (2013) Analyse und Design mit der UML 2.5 : objektorientierte Softwareentwicklung
 - Sehr gründlich und sorgfältig
- Heinrich, Mairon (2014) Objektorientierte Systemanalyse
 - Ähnliche Tätigkeiten wie Oesterreich, aber leicht andere Zuordnung zu OOA/OOD
- Laughlin, Pollice, West (2007) Objektorientierte Analyse & design von Kopf bis Fuß
 - Lustig, vermischt aber sehr stark Analyse, Design und Programmierung

Vielen Dank
für Ihre Aufmerksamkeit!

Für spätere Fragen:



**PROCESS
CONTROL
SYSTEMS**

**PROCESS
SYSTEMS
ENGINEERING**