

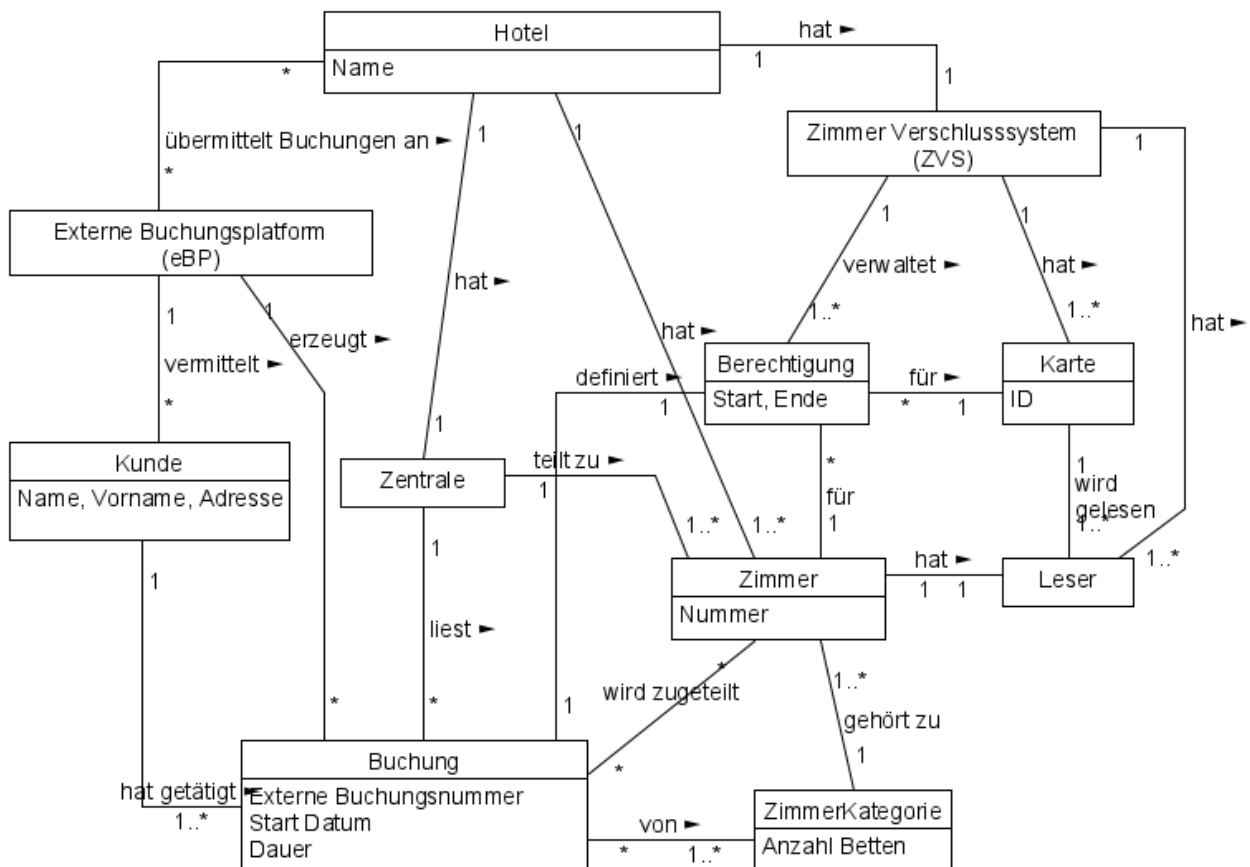
SWEN 1 SEP

Musterlösung

1 Anforderungsanalyse: Hotelsystem (42 P)

1.1 Einleitung und Problembeschreibung

1.2 Domänenmodell (DM) (16 P)



Bewertung

- Pro relevantes Konzept 1 P, max 9 P
- Für alle Attribute 3P
- Für alle Assoziationen 3P
- Referenzen zu Aufgabenstellung 1P
- Notationsfehler (z.B. gerichtete Assoziationen) werden direkt dort abgezogen

Paragraph 1: Hotel, Kunde (Attribute basieren auf allgemeinem Wissen)

Paragraph 2: externe Buchungsplattform (eBP), kein Check-In

Paragraph 3: Zimmer, Buchungsnummer (damit auch Buchung) vom eBP

Paragraph 4: ZVS, Karte (mit RFID), Leser(gerät), Berechtigung

Paragraph 5: Zimmerkategorie, Zentrale

Paragraph 6 Attribute Buchung

Referenzen können auch Nummern sein, die von Hand direkt in die Aufgabenstellung und ins Domänenmodell eingefügt werden.

1.3 Anwendungfall "UC01 – Check-In durchführen" (12 P)

Primärakteur: Gast (Kunde), der gerade eingetroffen ist **(1P)**

Stakeholders und Interessen (+ Vor/Nachbedingung 1P)

- Hotelgast: Möchte, dass Check-In korrekt und möglichst schnell abläuft
- Hotel Wie Gast + ohne Intervention Zentrale

Vorbedingung: Gast ist vor dem Terminal, Terminal läuft

Nachbedingung: Gast ist eingebucht und seine Zimmerkarten sind erfasst und konfiguriert.

Standardablauf (6P)

1. Gast gibt Buchungsnummer ein
2. System überprüft Buchungsnummer und zeigt die damit verbundenen Personendaten an
3. Gast bestätigt Korrektheit
4. Gast nimmt eine Zimmerkarte vom Stapel und führt sie dem Lesegerät beim Terminal zu.
5. System erhält ID der Zimmerkarte vom ZVS (oder direkt vom Leser). Darauf werden mit den Buchungsinformationen die Zimmer für diese Buchung reserviert und diese Information mit der Buchungsdauer dem ZVS mitgeteilt.
6. Schritte 4 und 5 werden solange wiederholt, bis alle Zimmer zugewiesen wurden
7. Das Verschlusssystem bestätigt die neue Konfiguration.
8. Das System teilt dem Kunden die Zimmernummer(n) mit und dass alles korrekt abgelaufen ist.

Erweiterungen (je 1P, maximal 4 P)

2a Unbekannte Buchungsnummer

1. System findet keine Buchung zur eingegebenen Nummer, teilt dies dem Gast mit und fährt bei Schritt 1 weiter.

3a Inkorrekte Daten

1. Der Gast stellt fest, dass die Buchungsdaten fehlerhaft sind. Er ruft mit dem vorhandenen Telefon die Zentrale an, um die Sache zu bereinigen
2. Zentrale nimmt Korrekturen an den Buchungsdaten vor und erlaubt es dann dem Gast, bei Schritt 4 weiterzufahren

5a ID der Zimmerkarte ist unbekannt.

1. System kennt die ID der Karte nicht. Es fordert den Gast auf, eine neue Karte vom Stapel zu nehmen.
2. Gast fährt bei Schritt 3 weiter.

5b Kein Zimmer verfügbar

1. System kann mit einfachem Algorithmus kein freies Zimmer gemäss den Buchungsdaten finden. Es fordert den Gast auf, mit dem Telefon die Zentrale anzurufen und das Problem so zu lösen
2. Zentrale weist von Hand ein Zimmer zu (vermutlich in einer besseren Kategorie)
3. Gast fährt bei Schritt 3 weiter.

5c ZVS kann nicht angesprochen werden

1. System teilt dem Gast den Fehler mit, alarmiert die Zentrale und informiert den Gast (respektive die nächsten Gäste), dass im Moment kein Check-In möglich ist.
2. Zentrale versucht Problem zu beheben und schaltet das System wieder aktiv, wenn dies geschehen ist.

- Akteur nicht benannt: -1P

- Passive Sätze: -1P

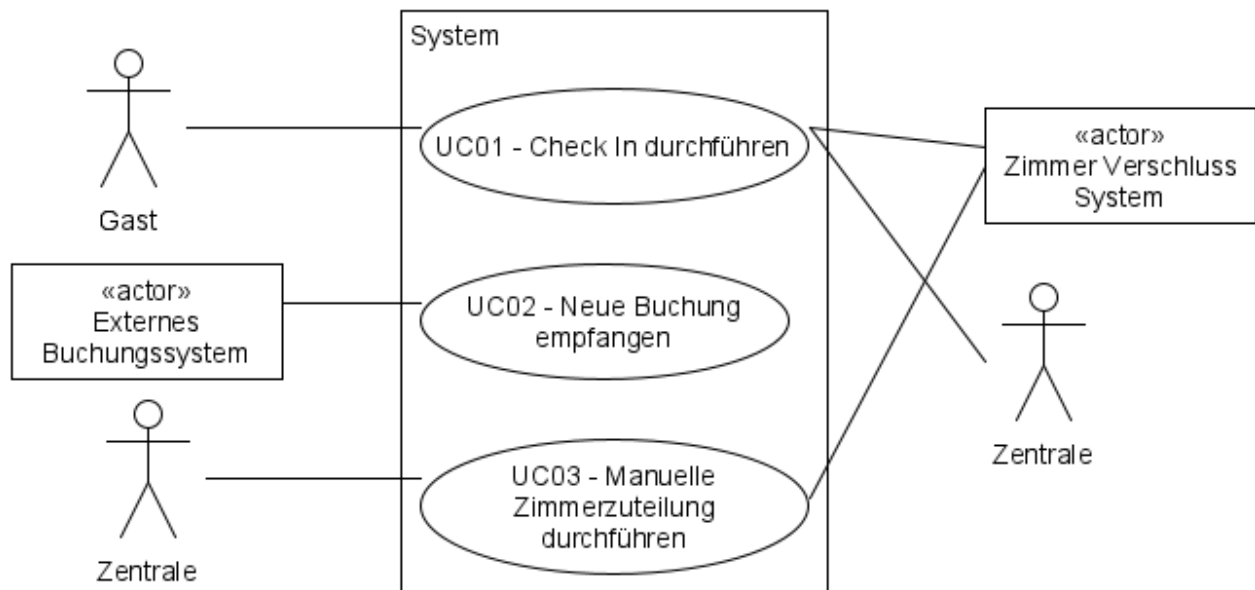
- Keine explizite System-Antwort: keine entsprechenden Punkte (= -3P)

1.4 Weitere Anwendungsfälle (4P)

- **Neue Buchung.** Externes Buchungssystem übermittelt dem System einen neue Buchung.(Paragraph 5)
- **Manuelle Zimmerzuteilung.** Zentrale ändert Buchungsdaten und weist ein Zimmer von Hand zu (Paragraph 5)
- **Reaktivierung des Check-In.** Zentrale kann nach einem Fehler im ZVS den Check-In wieder zulassen, nachdem die Reparatur erfolgt ist.

2 Anwendungsfälle sind erforderlich, pro Anwendungsfall 2P, maximal 4P

1.5 Anwendungsfalldiagramm (3P)



Pro Anwendungsfall, inklusive Akteure, **1P**

Keine Systemgrenzen: **-1**

1.6 Benutzerfreundlichkeit (3 P)

Paragraph 1: junge Leute → vermutlich guter Umgang mit Smartphones und Computer (**0.5 P**)

Paragraph 6: Grosser Bildschirm, Touch-Screen ohne Tastatur und Maus (**0.5 P**)

Hohe Benutzerfreundlichkeit ist so oder so bei einem System mit Publikumsbetrieb notwendig. Hier soll die Benutzerführung so einfach sein, dass die Bedienung ohne externe Hilfe (== teuer) auskommt.

Mögliches messbares (**1P**) Kriterium: Durchschnittlich sollte ein Check-In in 2 Min erledigt werden können, wenn eine Person das erste Mal dies macht.

Grosser Bildschirm: Grosse Bedienelemente und immer genügend Beschreibungen, was im Moment und nachfolgend zu tun ist. Keine kleinen Fonts, die auf dem grossen Bildschirm vermutlich schlecht lesbar sind.

Punkte:

- Referenz zu Paragraph: Je 0.5 Punkte
- Messbares Kriterium: 1 P
- Weitere Bemerkungen: **1 P**

1.7 Architekturanalyse (4 P)

Das externe System ZVS scheint in der ganzen Hotelkette dasselbe zu sein. Deshalb ist es streng genommen kein Variationspunkt, da ja keine Änderungen vorgesehen sind. Auch die Bemerkung des „Hauslieferanten“ deutet in diese Richtung. **(2 P)**

Als verantwortungsvoller Software Architekt jedoch ist es sinnvoll, eine Beurteilung vorzunehmen, ob das neue System wirklich sehr eng mit dem aktuellen ZVS verknüpft werden soll. Dabei stehen 2 Punkte im Vordergrund

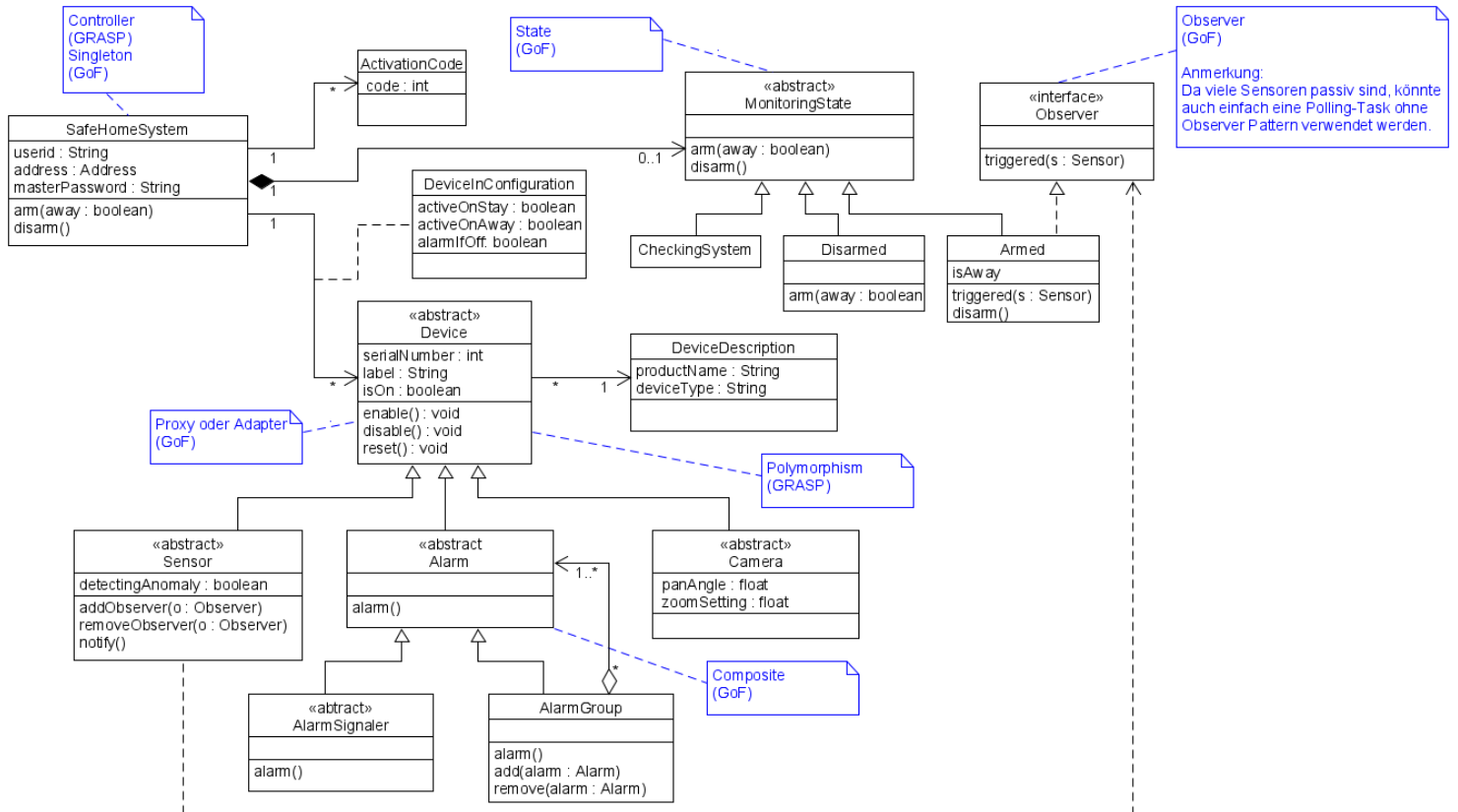
1. Es ist ja nicht nur der Hersteller, der ändern kann, sondern auch die technischen Spezifikationen der Schnittstelle
2. Eine Entkoppelung über Adapter ist voraussichtlich einfach zu realisieren

Aus diesen Gründen sollte dies als Entwicklungspunkt angesehen werden und trotz fehlender Anforderungen diesbezüglich eine Adaptionsschicht vorgesehen werden. **(2 P)**

2 Design: SafeHome (48 P)

2.1 Einleitung und Problemstellung

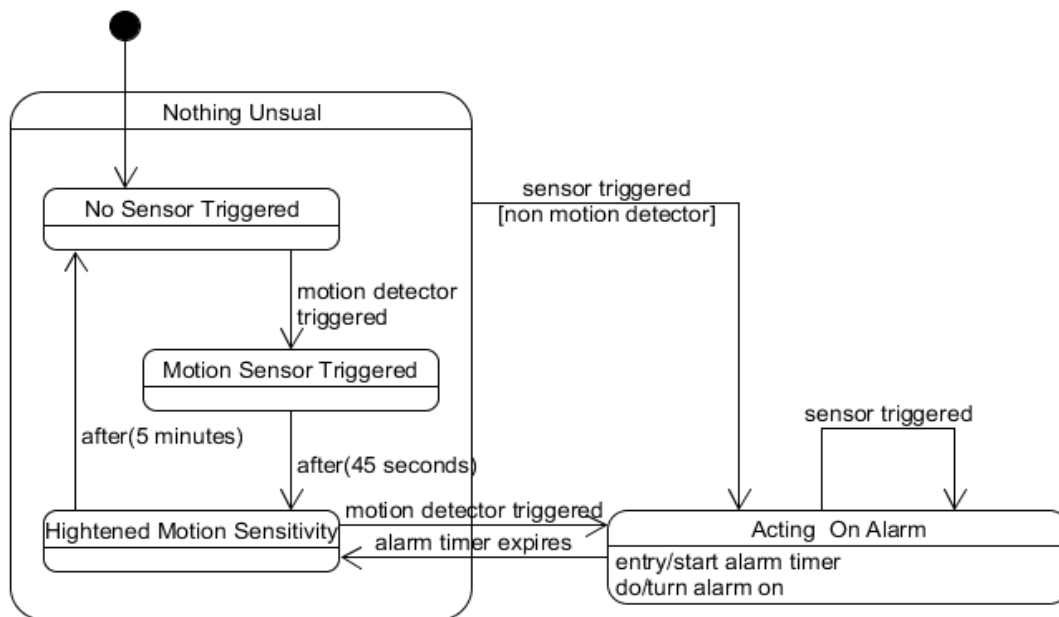
2.2 Design-Klassendiagramm (16 P)



Bewertung DCD:

- Controller SafeHomeSystem (2 P)
- Device und Subklassen mit Polymorphismus (2 P)
- Konfigurationsklasse mit Attributen für ein Device (4 P)
- Überwachung mit State Pattern (2 P)
- Sensorenabfrage mit Observer Pattern oder Polling (2 P)
- Alarm mit Composite Pattern (2 P)
- Begründung der Pattern (2 P)

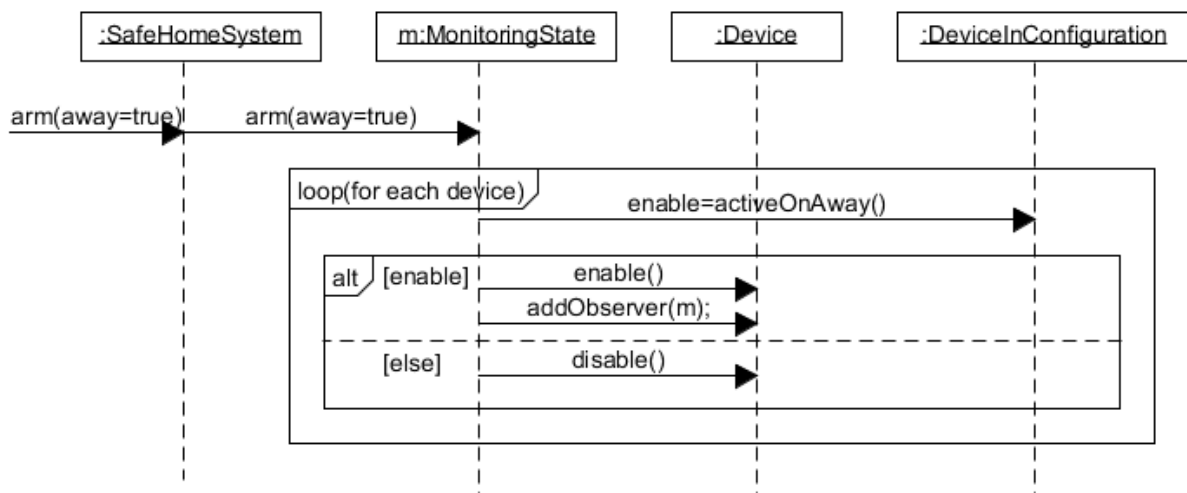
2.3 Verhalten während der Systemüberwachung (8P)



Bewertung:

- Für jeden korrekten Zustand und Transitionen 2 Punkte (max. 8 P)

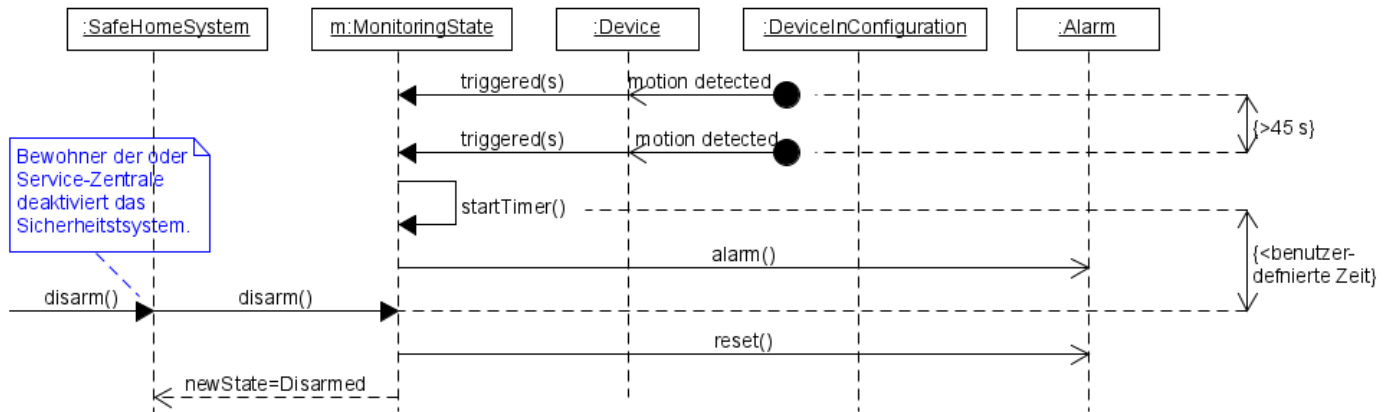
2.4 Systemüberwachung aktivieren (4 P)



Bewertung SD:

- Szenario Aktivierung der Systemüberwachung:
 - Alle Lifelines korrekt und mit DCD stimmig (2 P)
 - Ablauf, Messages und Timing korrekt (2 P)

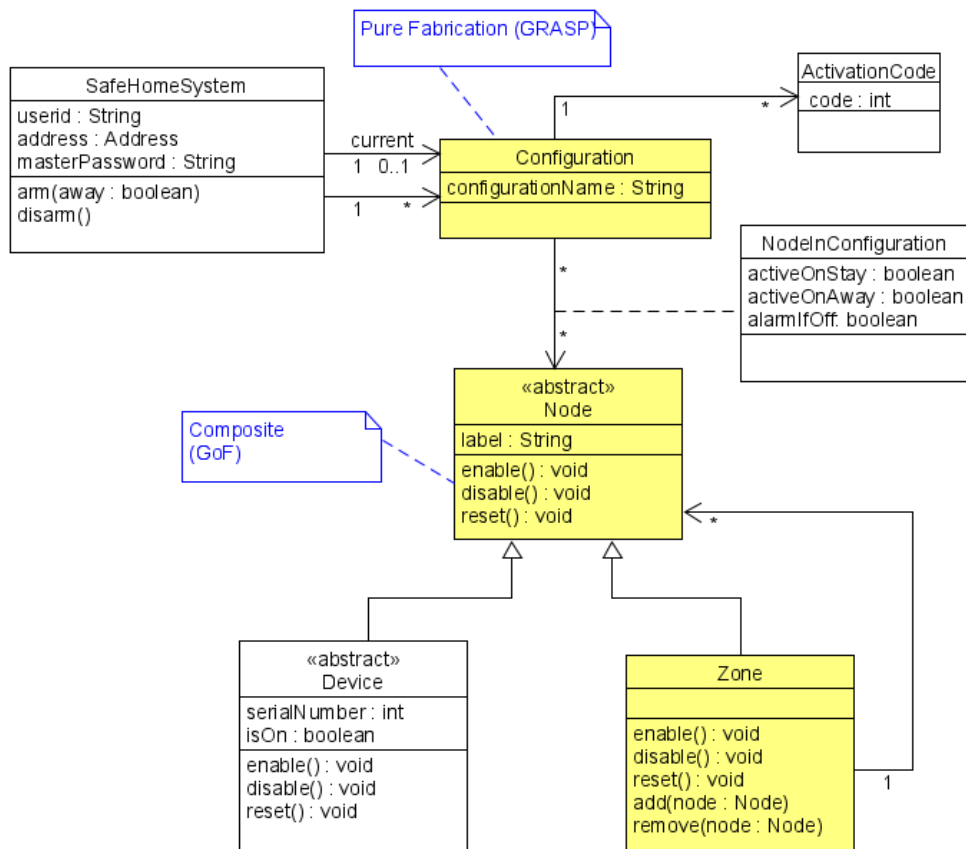
2.5 Szenario Einbrecher detektiert (6 P)



Bewertung SD:

- Szenario Motion Sensor triggered korrekt:
 - Alle Lifelines korrekt und mit DCD stimmig (2 P)
 - Ablauf, Messages und Timing korrekt (4)

2.6 Zusätzliche Anforderungen (10 P)



Bewertung:

- Klasse Konfiguration:
 - Klasse mit Attributen (2 P)
 - Assoziationen korrekt (2 P)
 - Begründung (1 P)
- Zonen mit Composite Pattern:
 - Klassen mit Attributen (2 P)
 - Assoziationen korrekt (2 P)
 - Begründung (1 P)

2.7 Fehlerszenarien (4 P)

Device – Sensor oder Aktor - defekt:

- Löst Falsch-Alarm aus (2 P): Manuelles Ein- und Ausschalten eines Gerätes muss möglich sein, um ein defektes Gerät ausschalten und ersetzen zu können. Kann evtl. Vorgängig mit einer Test-Funktion ermittelt und gelöst werden.
- Spricht nicht an bzw. ist tot (2 P): Watchdog bzw. periodische Überprüfung, ob ein Gerät noch verbunden ist. Falls es nicht mehr verbunden ist, sollte es so konfiguriert werden können, dass Alarm ausgelöst wird (evtl. von Einbrecher deaktiviert worden). Weiter sollte eine Test-Funktion hinzugefügt werden, die der Hausbewohner vor der Aktivierung oder periodisch ausführen könnte.

Es gibt noch viele weitere Fehlermöglichkeiten, aber in Bezug auf ein Sicherheitssystem sind natürlich defekte Sensoren und Aktoren die kritischen Komponenten ;-)