

AB045-Uebung Sequenzdiagramm-leer

Freitag, 29. Januar 2021 12:05

Dynamic Host Configuration Protocol

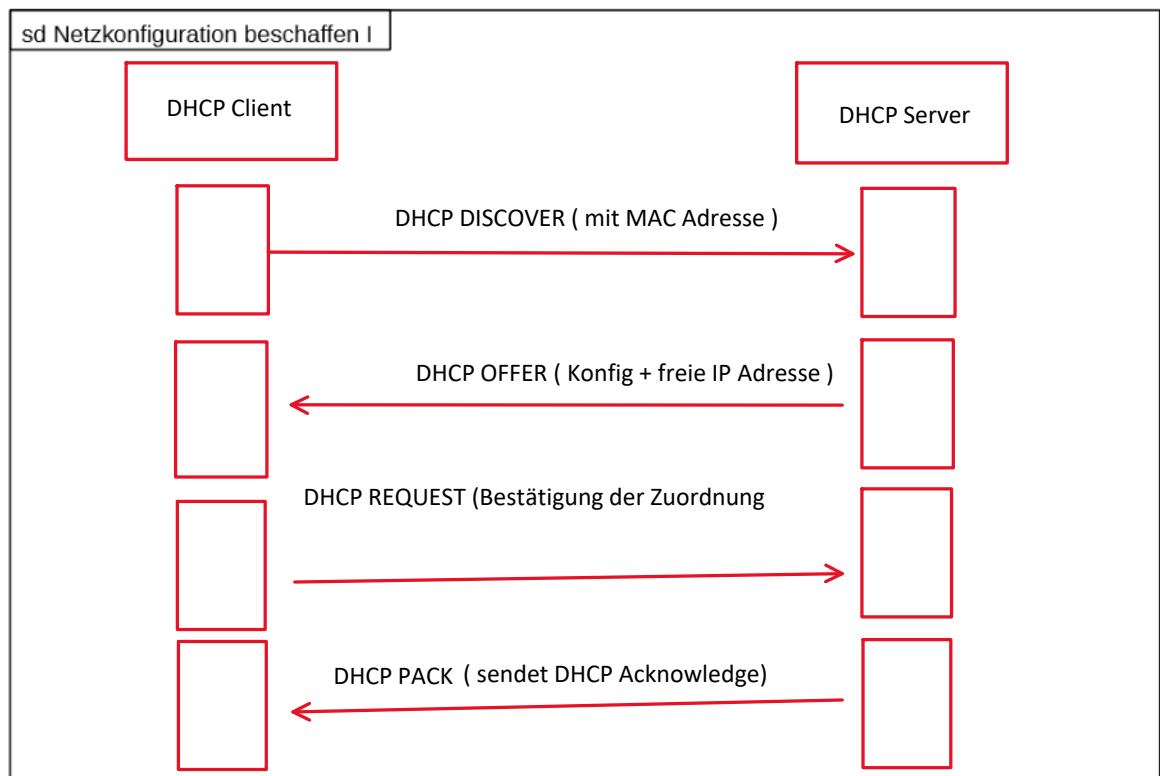
Aufgabe 1: Das *Dynamic Host Configuration Protocol* (DHCP) dient der automatischen Beschaffung von nötigen Netzwerkinformationen für Clients von DHCP-Servern. Diese Informationen umfassen u. a. eine IP-Adresse.

Die nachfolgende Auflistung beschreibt – im Sinne der Aufgabenstellen nur auszugsweise und vereinfacht! - wie Client und Server miteinander kommunizieren, um eine Netzwerkkonfiguration zu erhalten:

1. Der Client sendet eine Nachricht an alle DHCP-Server im Netzsegment (*DHCP_Discover*). Da der Client noch über keine IP-Adresse verfügt, dient seine MAC-Adresse als primäres Identifikationsmerkmal. Die Nachricht wird standardmäßig über UDP-Port 68 (clientseitig) und UDP-Port 67 (serverseitig) versandt.
2. Alle DHCP-Server die diese Nachricht erhalten, prüfen, ob sie über freie IP-Adressen verfügen. Ist dies der Fall antworten Sie dem Client mithilfe einer *DHCP_Offer*-Nachricht unter Angabe der Konfiguration (inkl. freier IP-Adresse) und ihrer Server-Id.
3. Der Client wählt aus den Angeboten der DHCP-Server eines aus und antwortet allen Server mit einer *DHCP-Request*-Nachricht, die als Parameter neben der gewählten Konfiguration, auch die Id des dazugehörigen Servers und seine (Client) MAC-Adresse enthält.
Alle DHCP-Server die diese Nachricht empfangen und deren Id *nicht* der im Datenpaket entspricht, werfen das Paket und betrachten die Client-Anfrage nach der Netzwerkkonfiguration als obsolet.
4. Der DHCP-Server, dessen Id mit der im Datenpaket übereinstimmt, bestätigt die Zuordnung der Konfiguration zum Client, indem er diesem eine *DHCP-Acknowledge*-Nachricht zusendet. Diese beinhaltet noch einmal die Konfiguration und seine Server-Id.
5. Der Client konfiguriert anhand der bestätigten Konfiguration seine Netzwerkschnittstelle und ist damit Teil des Netzwerks.

Entwerfen Sie für den beschriebenen Ablauf ein Sequenzdiagramm. Beschränken Sie sich dabei wie folgt:

- es gibt nur einen DHCP-Server im Netzsegment
- statt des Broadcasts kann eine direkte Kommunikation zum Server stattfinden





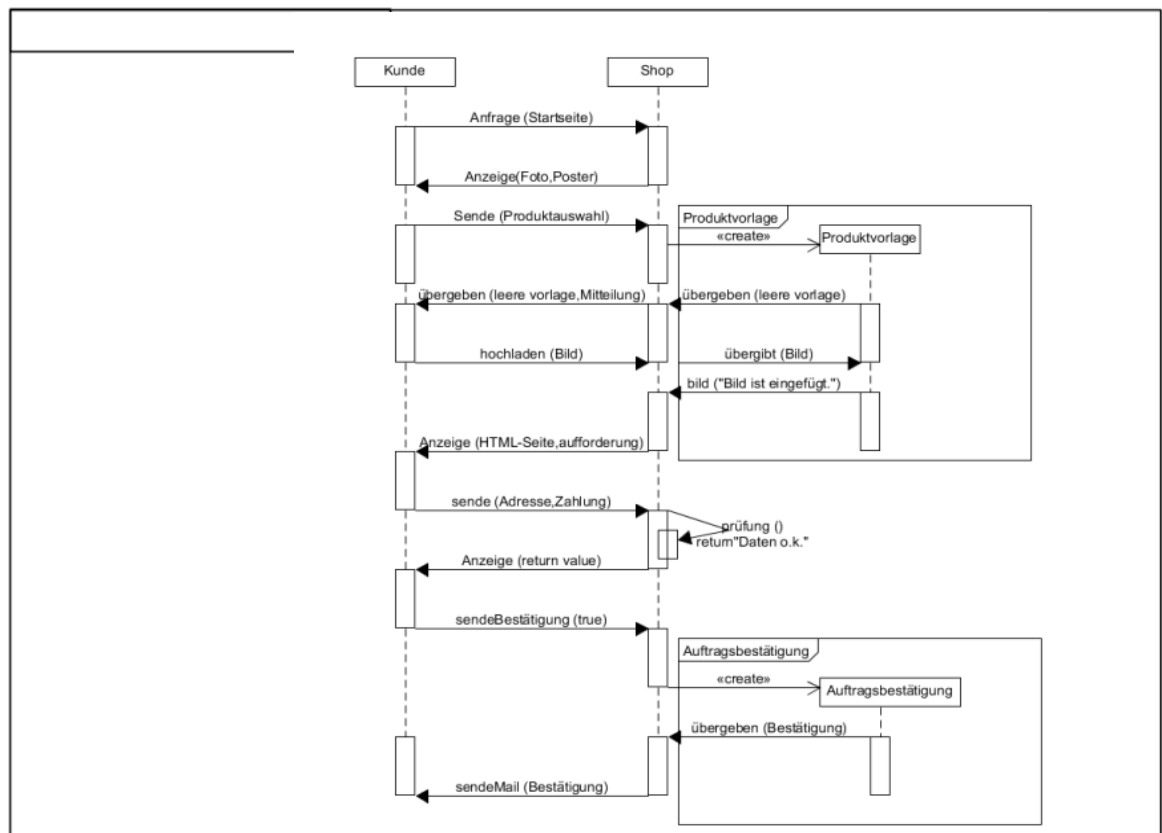
AB045-Ueb
ung Sequ...

Postershop

(Quelle: Abschlussprüfung FIAE, Sommer 2013, HS 2, Aufgabe a)

Aufgabe 2: Die X&Y IT GmbH verwendet zur Konstruktion und Dokumentation von Software- und IT-Systemen die UML. Der Ablauf einer Bestellung im Onlineshop der PixelPic AG soll in einem Sequenzdiagramm dargestellt werden. Der Ablauf wird wie folgt beschrieben:

- Kunde ruft die Startseite mit dem Angebot des Shops auf.
- Shop zeigt Kunden die Startseite mit den zur Wahl stehenden Produkten (Foto, Poster).
- Kunde teilt Shop Produktauswahl mit.
- Shop erstellt Produktvorlagen-Objekt; Aufruf der Funktion **create()** der entsprechenden Produktvorlagen-Klasse.
- Shop zeigt dem Kunden auf HTML-Seite die leere Produktvorlage und die Mitteilung, dass das Bild hochgeladen werden kann.
- Kunde lädt Bild hoch.
- Shop übergibt Bild an Produktvorlage; Aufruf der Funktion **bild()** des Produktvorlagen-Objekts.
- Produktvorlage meldet „Bild ist eingefügt“.
- Shop zeigt dem Kunden HTML-Seite mit fertiger Produktvorlage und fordert Kunden zur Eingabe der Adress- und Zahlungsdaten auf.
- Kunde übermittelt die Daten.
- Shop ruft die eigene Funktion **pruefung()** zur Prüfung der Daten auf.
- Funktion **pruefung()** gibt „Daten o. k.“ zurück. Der Fall „Daten nicht o. k.“ wird nicht betrachtet.
- Shop teilt Kunden mit, dass er zur Annahme des Auftrags bereit ist.
- Kunde erteilt Auftrag.
- Shop erstellt ein Auftragsbestätigungs-Objekt mit dem Aufruf der Funktion **create()** der Klasse Auftragsbestätigung.
- Shop verschickt Auftragsbestätigung per E-Mail an den Kunden.



AB067-Uebung Anwendungsfalldiagramm I

Montag, 8. Februar 2021 10:10



AB067-Ueb
ung Anwe...

Aufgabe 1

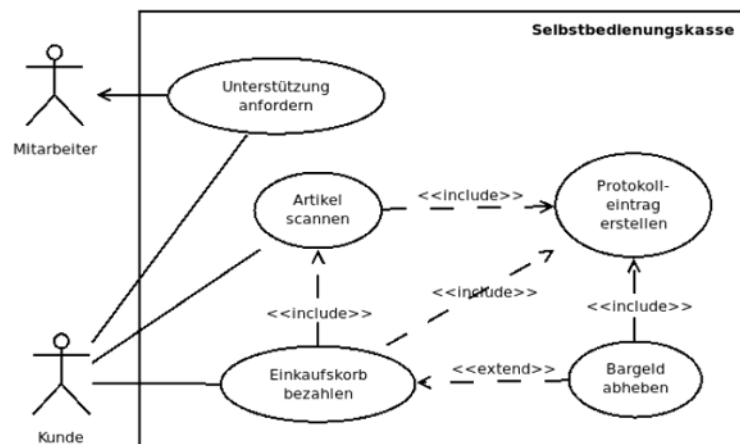
Sie sollen bei einem Kunden eine neue Kalenderanwendung einführen. Bei einem ersten Gespräch notieren Sie folgende Anforderungen:

<p>Ein Nutzer soll folgende Anwendungsfälle durchführen können:</p> <ul style="list-style-type: none"> • Termin anlegen • Termin ansehen • Termin ändern • Termin löschen • Kalender aktualisieren • Veränderungen am Kalender müssen eine automatische Aktualisierung nach sich ziehen 	<p>Ein Administrator soll folgende Anwendungsfälle durchführen können:</p> <ul style="list-style-type: none"> • Nutzer verwalten • Ein Administrator soll über die gleichen Rechte wie ein Nutzer verfügen
---	--

Erstellen Sie ein UML-Anwendungsfalldiagramm, welches die Anforderungen abbildet.

Aufgabe 2

Betrachten Sie das nachstehende UML-Anwendungsfalldiagramm. Erstellen Sie eine textuelle Beschreibung, welche die dargestellten Sachverhalte möglichst präzise wiedergibt.

**Aufgabe 3**

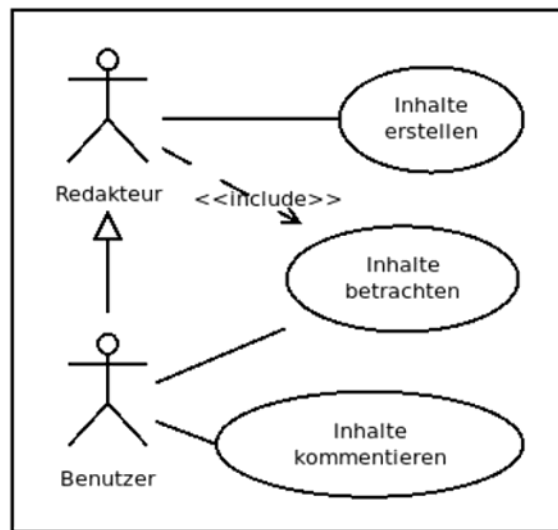
Sie arbeiten an der Neuentwicklung einer Kino-Website. Ein Nutzer der neuen Seite soll folgende Anwendungsfälle durchführen können:

- Kinoprogramm einsehen
- sich auf der Website einloggen
- Kinokarte reservieren
- Kinokartenkauf mit verschiedenen Zahlungsarten (Überweisung, Kreditkarte) durchführen
- Der Kauf von Karten ist nur nach Login möglich
- Kartenreservierungen sollen von der Website auf das Faxgerät des Betreibers gemeldet werden

Erstellen Sie ein UML-Anwendungsfalldiagramm, welches die Anforderungen abbildet.

Aufgabe 4

Betrachten Sie nachstehendes UML-Anwendungsfalldiagramm. Finden und markieren Sie die Fehler. Erstellen Sie eine formal korrekte Version des Diagramms.

**Aufgabe 5**

Sie erstellen die ersten Anwendungsfälle für ein Computerspiel. Folgende Funktionalität soll modelliert werden:

- Es gibt (von Menschen gesteuerte) Spieler-Charaktere („Spieler“) und (computergesteuerte) Nicht-Spieler-Charaktere („NSC“), die auf die Spielwelt einwirken.
- NSC bieten Spielern „Kampf-“, „Sammel-“ und „Transportquests“ in Abhängigkeit der Spieler-Eigenschaften an (z. B. Level, Fraktionszugehörigkeit), d. h. eine bestimmte Quest wird pro Spieler angeboten.
- Ein Spieler kann eine Quest durchführen. Dies setzt jedoch eine Questannahme zwingend voraus.
- Ein Spieler kann einen anderen Spieler zu einem Kampf herausfordern. Der herausgeforderte Spieler erhält einen entsprechenden Hinweis.

Erstellen Sie ein UML-Anwendungsfalldiagramm, welches die Anforderungen abbildet.



AB063-UML
-Steckbrie...

Steckbrief „Anwendungsfalldiagramm“

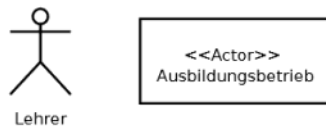
(Use Case Diagramm)

1. Verwendung

- modellieren Anwendungsfälle (Funktionalitäten) eines Systems aus Sicht externer Akteure _____
- hohes Abstraktionsniveau (Black-Box-Sicht) _____
- Darstellung von Anwendungsfällen, die ein externer Akteur wahrnehmen und nutzen kann _____

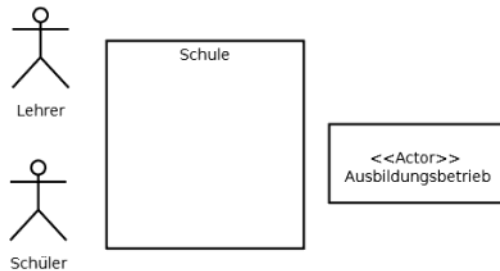
2. Aufbau

2.1 Akteure



- Akteure modellieren Typen oder Rollen, die ein externer Benutzer einnimmt _____
- Akteure sind Menschen oder Systeme _____

2.2 System

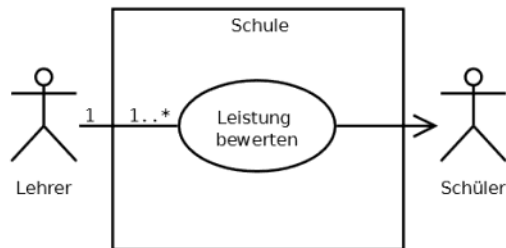


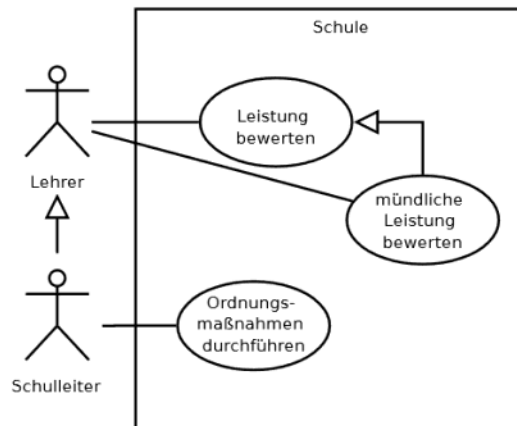
- Systeme werden als Rechteck dargestellt
- Systemname zwingend
- Rechteck = Systemgrenze
- Akteure befinden sich außerhalb des Systems

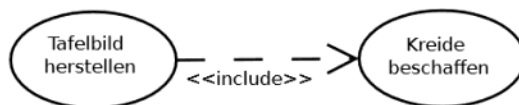
2.3 Anwendungsfall

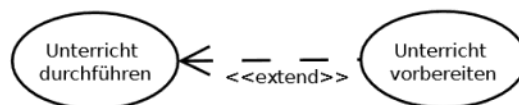


- Anwendungsfälle modellieren extern verfügbare Funktionalitäten eines Systems
- Das „was“ steht im Vordergrund _____

2.4 Assoziation

2.5 Generalisierung

2.6 „include“-Beziehung

2.7 „extend“-Beziehung



AB072-Ueb
ung-UML-...

AB072 - Übung UML-Aktivitätsdiagramm I

Version 1.0

Aufgabe 1

Sie sollen den Ablauf einer Neukundenberatung für die (fiktive) ActiveSales Leipzig GmbH (ASL) als UML-Aktivitätsdiagramm modellieren. Der Ablauf wird von ASL wie folgt beschrieben:

- Ein Vertriebsmitarbeiter ruft den Neukunden an und vereinbart einen Beratungstermin.
- Findet der Termin bei ASL statt, bereitet ein technischer Mitarbeiter (andere Person) einen Konferenzraum vor, andernfalls bereitet ein Berater einen Laptop vor (den er mit zum Kundentermin nimmt).
- Der Berater trifft sich mit dem Kunden.
- Anschließend sendet der Vertriebsmitarbeiter eine „Follow-up“-E-Mail an den Kunden.
- Enthält diese Nachverfolgungs-E-Mail konkreten Handlungsbedarf, wird dieser vom Berater in ein entsprechendes Angebot gefasst und an den Kunden verschickt.

Modellieren Sie ein Aktivitätsdiagramm, das die o. g. Anforderungen abbildet. Ordnen Sie die jeweiligen Knoten passenden Aktivitätsbereichen (Zuständigkeiten, „Lanes“) zu. Das Diagramm soll in englischer Fachsprache designed werden.

Aufgabe 2

Modellieren Sie ein Aktivitätsdiagramm, das folgender Formel entspricht:

$$\sum_{i=1}^5 i$$

Aufgabe 3

Ein Versandhändler beauftragt Ihren Arbeitgeber mit der Modellierung seines Bestellvorgangs. Der Prozess läuft wie folgt ab:

- Der Kunde gibt eine Bestellung auf
- Der Händler empfängt die Bestellung und entscheidet anschließend, ob er diese annimmt oder ablehnt. Lehnt er die Bestellung ab, wird diese sofort geschlossen und der Prozess ist beendet. Andernfalls wird die Bestellung weiterbearbeitet.
- Der Händler besorgt die bestellten Teile.
- Anschließend versendet er die Teile und – zeitgleich – die Rechnung.
- Der Kunde bezahlt die Rechnung.
- Der Händler prüft die Zahlung und schließt die Bestellung.

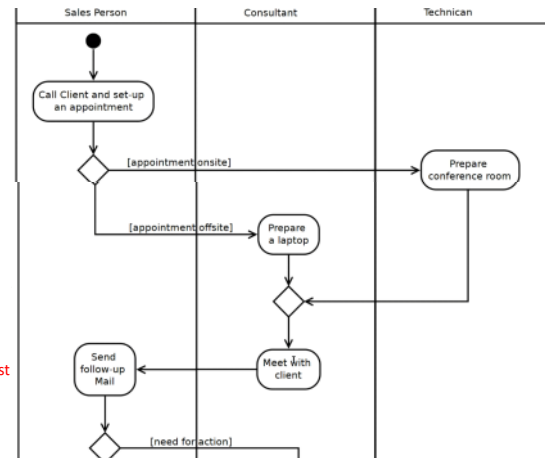
Modellieren Sie unter Verwendung von Aktivitätsbereichen (Zuständigkeiten, „Lanes“) den beschriebenen Prozess als Aktivitätsdiagramm.

Aufgabe 4

Der in Aufgabe 3 beschriebene Prozess soll so erweitert werden, dass ein Kunde eine Stornierungsanfrage durchführen kann, während der Händler die bestellten Teile besorgt. Diese Anfrage ist eine Unterbrechung des normalen Prozessablaufes und hat in jedem Fall die Stornierung und anschließende Beendigung des Prozesses der Bestellung zur Folge.

- Recherchieren Sie das Konzept der Unterbrechungen im UML-Aktivitätsdiagramm.
- Erweitern Sie das Diagramm aus Aufgabe 3 um die oben beschriebene Unterbrechung.

Follow Up ist





28.05.2021
_Schnee...

AB075 - UML - Zustandsdiagramm
Version 1.0

Steckbrief „Zustandsdiagramm“


(State Machine Diagram)

1. Verwendung

- Modellierung des Verhaltens von Systemen (wie Aktivitätsdiagramm)
- Zustandsdiagramme helfen während der Entwurfsphase Verhalten zu planen und ergänzen häufig Klassendiagramme („Objekt-Lebenszyklus“)
- Fokus liegt auf Reaktionen eines Systems (vgl. Aktivitätsdiagramm: Aktionen)
- im Besonderen zur Modellierung von Benutzeroberflächen (Verhaltens-Zustandsdiagramme) und Kommunikationsprotokollen (Protokoll-Zustandsdiagramme) geeignet

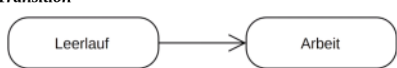
2. Aufbau

2.1 Zustand



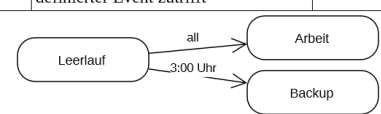
- Zustände definieren eine Situation, in der exakt definierte Bedingungen gelten
- Zustand in dem sich ein Objekt befindet wird als aktiv bezeichnet

2.2 Transition

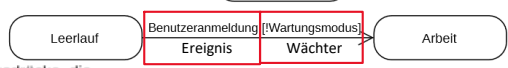


- Zustandsübergang:** überführt Objekt von einem Zustand in einen anderen
- gerichtete Assoziation

2.2.1 Ereignisse (Event)	Beschreibung	Beispiel
Call	Operation wird aufgerufen	Selfcheck()
Signal	(externes) Signal trifft ein	Benutzer berührt Touchscreen
Change	(interne) Attributwertänderung	Partnersystem jetzt online
Time	definierter Zeitpunkt erreicht	03:00 Uhr
Any-Receive	wird ausgeführt, wenn kein anderer definierter Event zutrifft	„all“ (fest definiert)




2.2.2 Wächter (Guards)



- Guards sind bool'sche Ausdrücke, die Transitionen nur bei „true“ zulassen
- Nicht verwechseln: Change-Events (beachte eckige Klammern)

2.2.3 Effekte (effects)



- Effekte beschreiben Aktionen, die während der Transition ausgeführt werden

2.2.4 Zusammenfassung:

<event> [<guard>] / <effect>

Sebastian Schneemann, BSZ7 Leipzig, s.schneemann@bsz7-leipzig.de
Seite 1

AB075 - UML - Zustandsdiagramm
Version 1.0




2.3 Zustände mit internen Aktionen

Berechnung
 Teilergebnis berechnet / Fortschrittsbalken aktualisieren

Arbeit
 entry / ToDo-Liste sichten
 do / ToDo-Liste abarbeiten
 exit / ToDo-Liste aktualisieren

- Interne Aktionen führen Funktionalität aus, ohne den Zustand zu wechseln
- Events können benutzerdefiniert oder vorgegeben sein (Standardaktionen)


2.4 Start- und Endzustand, Terminator

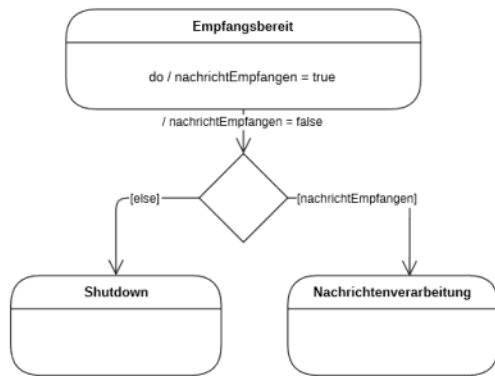
Startzustand
Endzustand
Terminator

- Startzustand (1)-flüchtig, ohne Wächter
- Endzustand n Stück
- Terminator = Zerstörung des zugrundeliegenden Objekts

2.5 Entscheidungen und Kreuzung



Der auf „dem Weg“ zur Entscheidung durch die

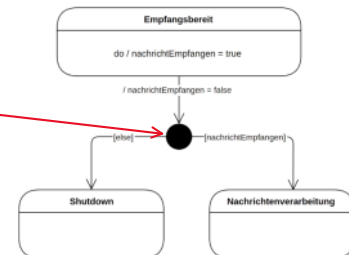


Der auf „dem Weg“ zur Entscheidung durch die Transition ausgeführte Effekt (nachrichtEmpfang=false) wird bei den Wächtern berücksichtigt

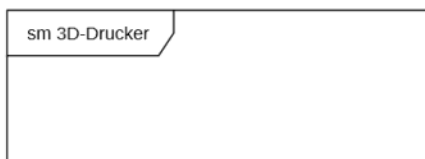
→ Zielzustand: Shutdown

- _____
- _____
- _____
- _____

- Der auf „dem Weg“ zur Kreuzung durch die Transition ausgeführte Effekt wird erst nach dem passieren der Wächter umgesetzt → Zielzustand: Nachrichtenverarbeitung



2.6 Rahmen



- Aufbau „<sm> <Name>“ (sm = state machine)
- Rahmen definiert Zustandsdiagramm
- kann Ein- und Austrittspunkte haben

2.7 Weitere Elemente im Zustandsdiagramm

- Zusammengesetzte Zustände zur Modellierung von Zustandshierarchien
- Regionen um zusammengesetzte Zustände und/oder Zustandsdiagramme in Teilmengen zu teilen
- Generalisierungen bzw. Spezialisierungen von Zustandsdiagrammen
- Selbsttransition (Selbstreferenzierungen) um darzustellen, dass Ziel- und Quellzustand identisch sind
- Signalsendung und -empfang