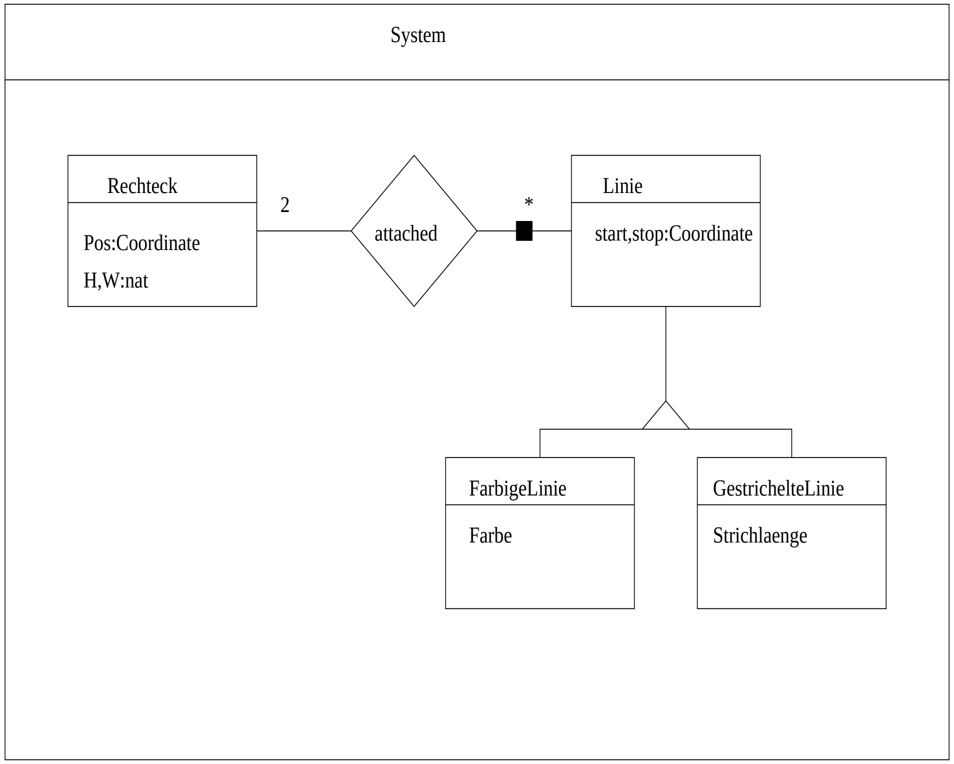
Aufgabe 1 (Fusion)

(Geschätzte Bearbeitungszeit: 35 min.)

1. Erstellen Sie ein Objektmodell für eine einfache Zeichenapplikation: Ein System besteht aus Rechtecken, die durch Linien verknüpft sind. Dabei verknüpft eine Linie immer genau zwei Rechtecke. Von einem Rechteck können beliebig viele Linien ausgehen. Größe und Position von Rechtecken und Linien sind durch geeignete Attribute festzulegen. Linien können schwarz oder farbig sein und durchgezogen oder gestrichelt dargestellt werden. *8 Punkte*

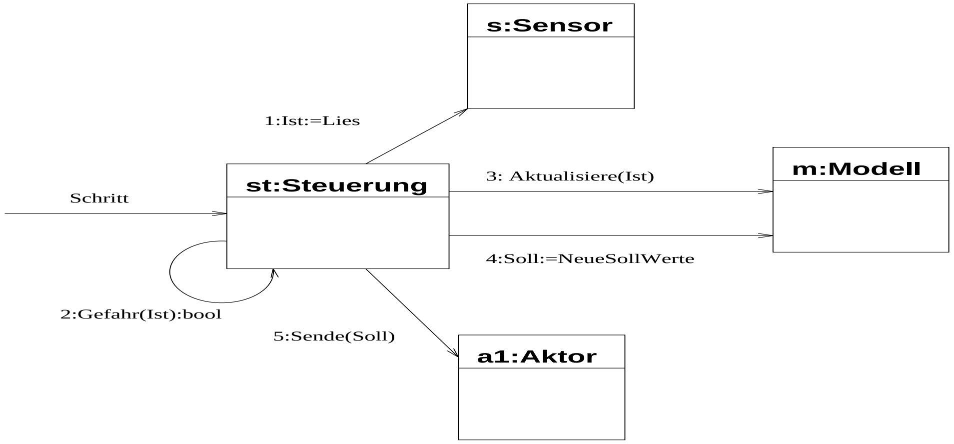
   Hinweis: Modellieren Sie nur die oben beschriebenen Informationen!

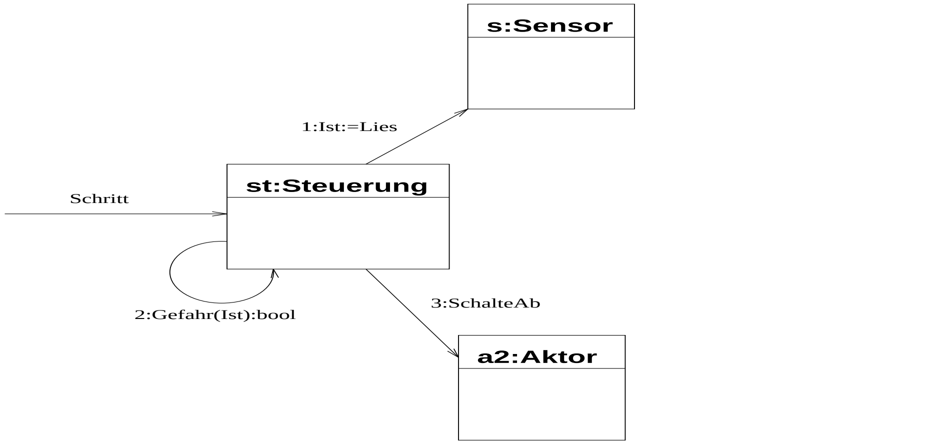


2. Entwickeln Sie (einen oder mehrere) Objektinteraktionsgraphen für den (zyklischen) Steuerschritt einer einfachen Prozeßsteuerung. In einem Steuerschritt sind Objekte aus folgenden Klassen involviert: eine $Steuerung$, ein $Sensor$, zwei $Aktoren$, und ein internes $Modell$. Das Steuerungsobjekt kümmert sich um die Gesamtkoordination, der Sensor liefert frisch gemessene Werte des Prozesses, durch einen Aktor kann auf den Prozeß eingewirkt werden und das interne Regelungsmodell des (physikalischen) Prozesses wird zur Berechnung neuer Stellgrößen verwendet. *8 Punkte*

   Die Nachbedingung der Methode $Schritt$ der Klasse $Steuerung$ lautet wie folgt: Das Steuerobjekt hat die IST-Werte von Sensor s gelesen. Falls die Werte eine Gefahrensituation anzeigen, ist über **Aktor** $a1$ eine Notabschaltung vorgenommen worden, andernfalls wurde das interne Regelungsmodell m mit den frischen Sensorwerten aktualisiert und an **Aktor** $a2$ wurden die aus dem internen Modell berechneten neuen SOLL-Werte übermittelt.

   Hinweis:
   Führen sie beim Erstellen des/der Graphen geeignete Methoden für die angebenen Klassen ein.
   Wählen Sie für diese Methoden geeignete Ein-/Ausgabeparameter.





3. Beschreiben Sie kurz (jeweils in 1-2 Sätzen) die verschiedenen Sichtbarkeitsbeziehungen und ihre Notation nach der Fusion-Methode. *4 Punkte*

- Lebensdauer einer Referenz
  Die Lebensdauer eines Servers bezüglich eines Klienten kann permanent oder dynamisch sein.
- Server-Sichtbarkeit
  Die Sichtbarkeit eines Servers bezüglich eines Klienten kann ausschließlich sein.
- Server-Bindung
  Die Bindung eines Servers bezüglich eines Klienten kann gebunden oder ungebunden sein.

- Änderbarkeit der Referenz
  Die Änderbarkeit der Referenz zwischen Server und Klient kann konstant oder variabel sein.

Aufgabe 2 (Z)

(Geschätzte Bearbeitungszeit: 35 min.)

1. Wir modellieren einen Getränke- oder Süßigkeitenautomaten, wie sie z.B. auf Bahnhöfen zu finden sind. Diese Automaten habe eine bestimmte Menge von Produkten im Angebot. Jedes der angebotenen Produkte ist in einer bestimmten Anzahl vorhanden und hat einen Preis. Diese werden als natürliche Zahl ausdrückt. Außerdem hat der Automat einen gewissen Geldbestand.

   Der Basistyp $Produkt$ bezeichnet die Produkte, die in Automaten angeboten werden können.

   $$[Produkt]$$

   Folgendes Zustandschema definiert die Zustände des Automaten:

   $$\begin{array}{|l}
   \textit{Automat} \\
   \hline
   Anzahl : Produkt \pfun \mathbb{N} \\
   Preis : Produkt \pfun \mathbb{N} \\
   Geldbestand : \mathbb{N} \\
   \hline
   \mathrm{dom}\ Anzahl = \mathrm{dom}\ Preis \\
   \forall\, prod : \mathrm{dom}\ Anzahl \bullet Anzahl\ prod \leq 50 \\
   \hline
   \end{array}$$

   Dabei ist $\mathrm{dom}\ Anzahl$ die Menge der Produkte, die in dem Automaten normalerweise vorhanden sind. Wenn ein Produkt $prod$ ausgegangen ist, gilt $Anzahl\ prod = 0$. Von jedem angebotenen Produkt passen höchstens 50 Stück in den Automaten.

   (a) Wenn der Automat neu in Betrieb genommen wird, hat er noch keine Anzahl von Produkten, und der Geldbestand ist 0. Spezifizieren Sie die Initialisierungsoperation $InitAutomat$, die dies ausdrückt. *2 Punkte*

   ### Lösung

   $$\begin{array}{|l}
   \textit{InitAutomat} \\
   \hline
   Automat' \\
   \hline
   Anzahl' = \emptyset \\
   Geldbestand' = 0 \\
   \hline
   \end{array}$$

   (b) Spezifizieren Sie eine Operation $Verkaufen$. Eingabe ist ein Produkt und ein Geldbetrag (eine natürliche Zahl). Ausgabe ist das Wechselgeld, das der Kunde zurückerhält, ausgedrückt als natürliche Zahl. Der Zustand des Automaten ist entsprechend zu aktualisieren. Sie können davon ausgehen, daß immer genügend Wechselgeld im Automaten vorhanden ist. Betrachten Sie nur den Normalfall! *4 Punkte*

   ### Lösung

   $$\begin{array}{|l}
   \textit{Verkaufen} \\
   \hline
   \Delta Automat \\
   P? : Produkt \\
   Einzahlung? : \mathbb{N} \\
   Wechselgeld! : \mathbb{N} \\
   \hline
   P? \in \mathrm{dom}\ Anzahl \\
   Anzahl\ P? > 0 \\
   Einzahlung? \geq Preis\ P? \\
   Anzahl' = Anzahl \oplus \{P? \mapsto Anzahl\ P? - 1\} \\
   Preis' = Preis \\
   Geldbestand' = Geldbestand + Preis\ P? \\
   Wechselgeld! = Einzahlung? - Preis\ P? \\
   \hline
   \end{array}$$

   (c) Spezifizieren Sie eine Operation *Auffüllen*. Der Automatenbetreiber kommt und füllt den Bestand des Automaten nach, so daß von allen angebotenen Produkten wieder 50 Stück vorhanden sind. Gleichzeitig leert er den Geldbestand und läßt nur 100 Geldeinheiten Wechselgeld zurück. Betrachten Sie nur den Normalfall! *4 Punkte*

   ### Lösung

   $$\begin{array}{|l}
   \textit{Auffüllen} \\
   \hline
   \Delta Automat \\
   \hline
   Anzahl' = \{P : \mathrm{dom}\ Anzahl \bullet P \mapsto 50\} \\
   Preis' = Preis \\
   Geldbestand' = 100 \\
   \hline
   \end{array}$$

(d) Spezifizieren Sie eine Operation $Preisabfrage$. Eingabe ist ein Produkt, Ausgabe ist der Preis des Produktes. Betrachten Sie nur den Normalfall! *2 Punkte*

Lösung

$$
\begin{array}{|l}
\multicolumn{1}{l}{\underline{\ Preisabfrage\ }} \\
\Xi Automat \\
P? : Produkt \\
Pr! : \mathbb{N} \\
\hline
P? \in \mathrm{dom}\ Anzahl \\
Pr! = Preis\ P? \\
\hline
\end{array}
$$

(e) Geben Sie für jede der in den Aufgabenteilen (b) – (d) definierten Operationen die Vorbedingung an. *2 Punkte*

Lösung

Operation	Vorbedingung
$InitAutomat$	$true$
$Verkaufen$	$P? \in \mathrm{dom}\ Anzahl \wedge Anzahl\ P? > 0 \wedge Einzahlung? \geq Preis\ P?$
	$true$
$Preisabfrage$	$P? \in \mathrm{dom}\ Anzahl$

2. Nennen Sie die Schritte, die man in Z durchführt, um eine benutzerfreundliche (d.h. totale) Systemoperation zu spezifizieren. Wie würden diese Schritte für die Operation $Preisabfrage$ aussehen? Verlangt ist nur eine Beschreibung der Schritte, nicht ihre Durchführung! *4 Punkte*

Lösung

(a) Spezifiziere die Operation für den Normalfall.

(b) Analysiere die Vorbedingung.

(c) Spezifiziere Operationen für die Fehlerfälle.

(d) Benutze den Schemakalkül, um die Operationen für den Normalfall und für die Fehlerfälle so zu kombinieren, daß eine Operation mit Vorbedingung $true$ entsteht.

Für die Operation $Preisabfrage$:

(a) Operation für den Normalfall bereits definiert.

(b) Vorbedingung bereits analysiert.

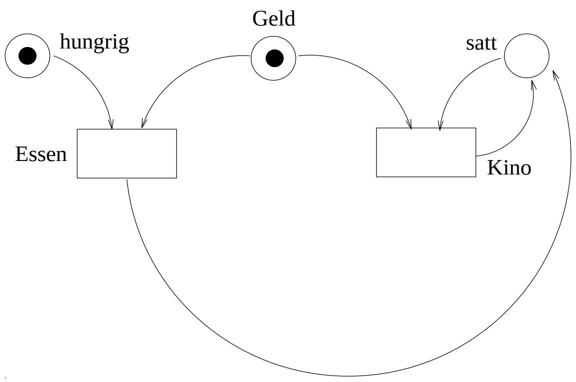
(c) Spezifiziere eine Operation $ProduktNichtImAngebot$, deren Vorbedingung $P? \notin \mathrm{dom}\ Angebot$ ist und die den Automaten nicht verändert ($\Xi Automat$).

(d) $PreisabfrageTotal \mathrel{\widehat{=}} Preisabfrage \vee ProduktNichtImAngebot$

Aufgabe 3 (Petri-Netze)

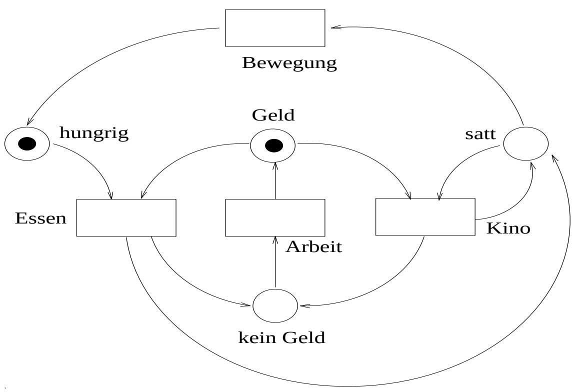
(Geschätzte Bearbeitungszeit: 25 min.)

1. Betrachten Sie folgendes Petri-Netz:



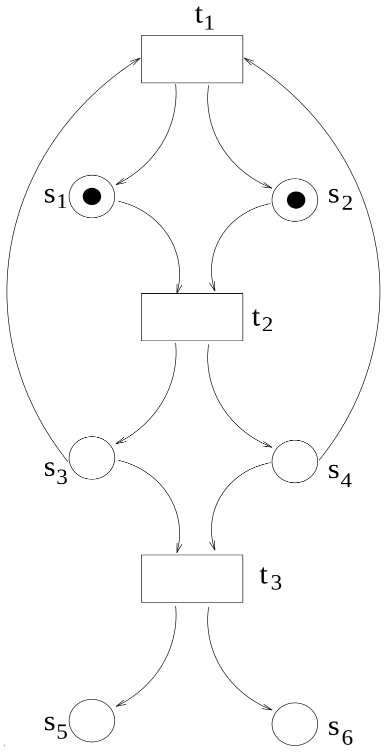
Erweitern Sie das Netz um eine Stelle $kein\_Geld$, als Ergebnis sowohl von $Kino$ als auch von $Essen$, sowie um die Transitionen $Arbeit$ (von $kein\_Geld$ nach $Geld$) und $Bewegung$ (von $satt$ nach $hungrig$). *3 Punkte*

Lösung



3 Punkte

Betrachten Sie folgendes Petri-Netz:



1. Welche der Transitionen können schalten? *1 Punkt*
2. Stellen Sie das Netz als Netzmatrix $\underline{N}$ und seine Anfangsmarkierung als Spaltenvektor $\underline{M}$ dar. *2 Punkte*
3. Führen Sie das Schalten einer zulässigen Transition als Vektoraddition durch. *1 Punkt*

Lösung

(a) Es kann genau Transition t_2 schalten.

(b) Netzmatrix: $\underline{N} = \begin{array}{c|ccc} & t_1 & t_2 & t_3 \\ \hline s_1 & 1 & -1 & 0 \\ s_2 & 1 & -1 & 0 \\ s_3 & -1 & 1 & -1 \\ s_4 & -1 & 1 & -1 \\ s_5 & 0 & 0 & 1 \end{array}$

$$\text{Anfangsmarkierung } \underline{M} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Schalten von Transition t_2:

$$M + t_2 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} -1 \\ -1 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

Welche Bedeutung haben Stellen, Transitionen und Markierungen in einem Stellen/Transitionsnetz, d.h. welche Dinge werden mit Stellen, Transitionen bzw. Markierungen modelliert? *3 Punkte*

Lösung

Stellen: passive Elemente des Systems, Zustände

Transitionen: aktive Elemente des Systems, die produzieren, transportieren oder verändern

Markierungen: Verteilung von Resourcen oder Objekten

Aufgabe 4 (Statecharts)

(Geschätzte Bearbeitungszeit: 25 min.)

1. In dieser Aufgabe sollen Sie die folgende, stark vereinfachte Benutzerschnittstelle eines Bankautomaten mit einem Statechart modellieren: Der Kunde führt zuerst seine Karte ein, muß daraufhin seine korrekte Geheimzahl eingeben (zwei Fehlversuche erlaubt, sonst Abbruch), und kann danach den gewünschten Betrag eingeben. Der Automaten stellt diesen Betrag zur Verfügung und gibt die Karte aus, die vom Kunden herausgenommen wird. Ist 60 Sekunden nach Einführung der Karte noch kein Auszahlungsbetrag eingegeben, dann bricht der Automat den Vorgang ab. *7 Punkte*

   Wichtig!: Verwenden sie bei der Modellierung nur die folgenden 5 Ereignisse, die den vom Kunden bei der Benutzung durchgeführten Tätigkeiten entsprechen!!!

   KE Karte wird eingeführt,

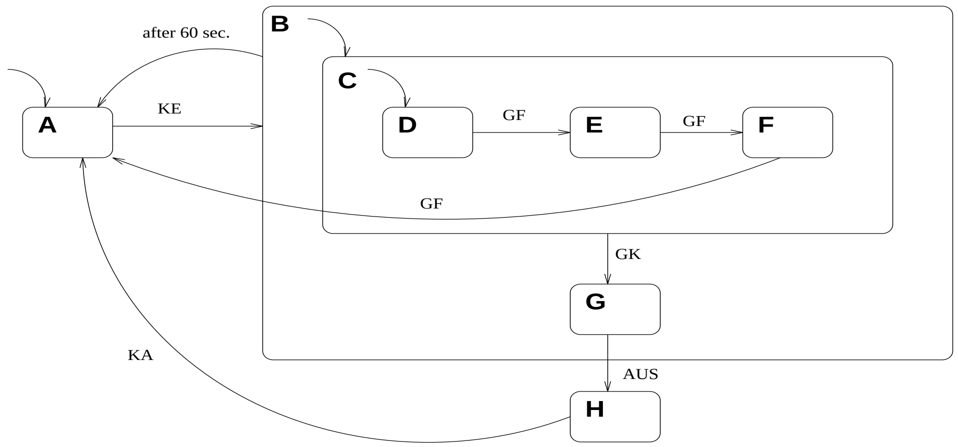
   KA Karte wird herausgenommen,

   GK Geheimzahl wird korrekt eingegeben,

   GF Geheimzahl wird falsch eingegeben,

   AUS Auszahlungsbetrag wird eingegeben,

   Hinweis: Modellieren Sie nur das oben beschriebene Verhalten des Automaten. Zusätzliche (Fehler-)Situationen brauchen nicht berücksichtigt zu werden.



2. Beschreiben Sie in 2 bis 3 Sätzen den Unterschied zwischen hierarchischer und paralleler Komposition und wofür diese Kompositionsmechanismen bei der Modellierung verwandt werden. *3 Punkte*

Bei der hierarchischen Komposition werden mehrere Zustände durch exklusive Disjunktion (XOR) miteinander zu einem neuen Oberzustand verknüppft.

Bei der parallelen Komposition werden mehrere Zustände durch Konjunktion (AND) miteinander verknüpft.

Hierarische Komposition wird zur schrittweisen Verfeinerung bei der Modellierung sequentieller Systemteile verwendet.

Parallele Komposition wird hingegen zur parallelen Verknüpfung von Systemteilen verwendet.