Technische Universität Berlin
Institut für Kommunikations- und Softwaretechnik
FG Softwaretechnik
Franklinstr. 28/29
10587 Berlin

Koch
Bittner

Musterlösung

Klausur

Softwaretechnik

25. Februar 1999

Aufgabe 1 (Fusion)

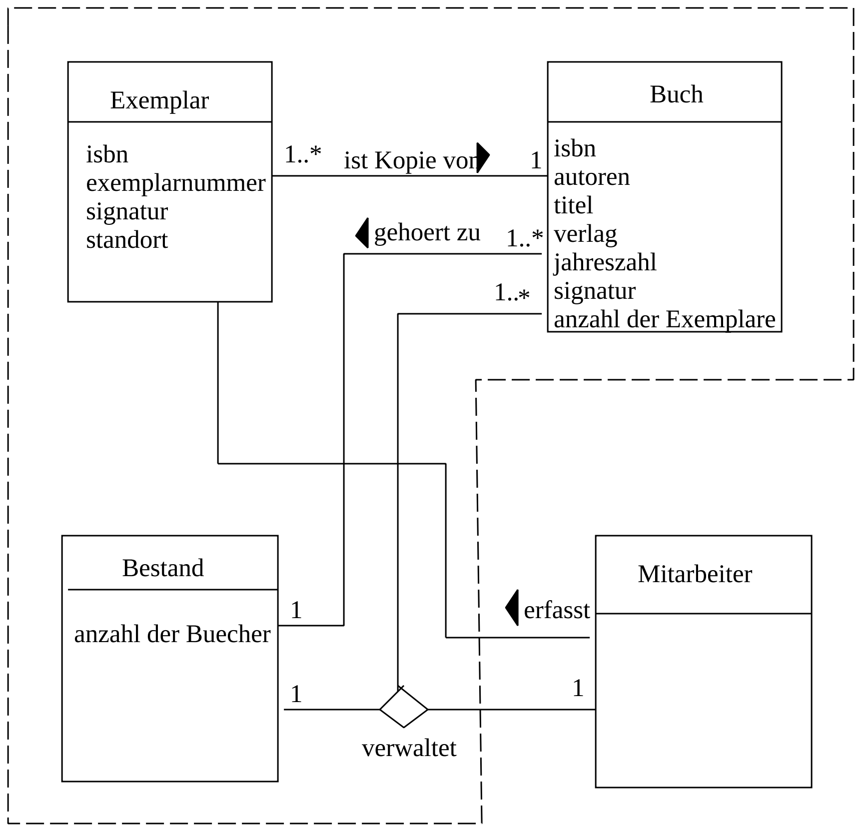
(Geschätzte Bearbeitungszeit: 30 min.)

Stadtbücherei

In einer Stadtbücherei werden ständig neue Bücher angeschafft.

Abhängig von der Nachfrage der Leser kann von einem Buch ein Exemplar oder mehrere Exemplare erworben werden. Verschiedene Exemplare eines Buches können an unterschiedlichen Standorten sein.

Das folgende Klassenmodell modelliert die Beziehung zwischen Klassen des Systems, wobei zu beachten ist, daß die Zuordnung von Exemplaren zu Büchern **eindeutig** ist und Buch (als Abstraktion) von den Exemplaren zu unterscheiden ist.



Die Analyse ergibt, daß es wenigstens die beiden folgenden Systemoperationen geben soll:

- *Aufnahme_Buch:*
  Es sollen die Daten eines Buches erfaßt und in den Bestand der Stadtbücherei aufgenommen werden.

- *Neuerwerb_Exemplar :*
  Es soll ein neues Exemplar eines bereits im Bestand befindlichen Buches der Stadtbücherei zugefügt werden.

Aufgabenstellung:

a.) Geben Sie an, welche Eingabeparameter die Systemoperation *Aufnahme_Buch* und *Neuerwerb_ Exemplar* mindestens haben sollten. *5 Punkte*

b.) Führen Sie alle Klassen auf, deren Objekte am Effekt der beiden Operationen beteiligt sind. *5 Punkte*

c.) Was ändert die Systemoperation *Neuerwerb_Exemplar* im System. Geben Sie die entsprechenden Teile des Operationschemas an. *20 Punkte*

d.) Zählen Sie auf, in welchen Modellen der Methode Fusion Klassen und in welchen Modellen Objekte (als Instanz von Klassen) zugrunde gelegt werden. *5 Punkte*

   Hinweis: Nur in Stichworten beschreiben, bitte keine Abhandlung!!!

zu a.) Aufnahme_Buch: isbn, autoren, titel, verlag, jahreszahl, signatur

alternativ: isbn, autoren, titel, verlag, jahreszahl
weil Signatur vom System vergeben werden soll.

Neuerwerb_Exemplar: isbn, signatur, standort

alternativ : isbn
weil standort und signatur vom System vergeben werden soll.

zu b.) Aufnahme_Buch: Bestand, Buch

Neuerwerb_Exemplar: Exemplar, Buch
alternativ: Bestand, Buch und Exemplar

zu c.) Operation:Neuerwerb_Exemplar

Reads: supplied isbn, signatur, standort

Changes: $\underline{\text{:Exemplar}}$\{new\}, $\underline{\text{:Buch}}$

Pre: $\exists$ $\underline{\text{:Buch}}$ | $\underline{\text{:Buch}}$.isbn = isbn

Post:
$\underline{\text{:Exemplar}}$.isbn = isbn
$\underline{\text{:Exemplar}}$.signatur = signatur
$\underline{\text{:Exemplar}}$.standort = standort
$\underline{\text{:Expemplar}}$.exemplarnummer = $\underline{\text{:Buch}}$.anzahlderExemplare'
$\underline{\text{:Buch}}$.anzahlderExemplare' = $\underline{\text{:Buch}}$.anzahlderExemplare+1

zu d.) Klassen: Klassenmodell, Systemklassenmodell, Referenzierungsmodell, Klassenschnittstelle
Objekte: Operationsmodell, Timeline Diagramme, Objektinteraktionsmodell

Aufgabe 2 (Object-Z)

(Geschätzte Bearbeitungszeit: 60 min.)

Schiffchen versenken

Auf einem Spielfeld von 6 x 6 Feldern können 3 "Schiffchen" markiert werden: jedes Schiffchen besteht aus 3 aufeinander folgenden Spielfeldern in waagerechter oder senkrechter Richtung.

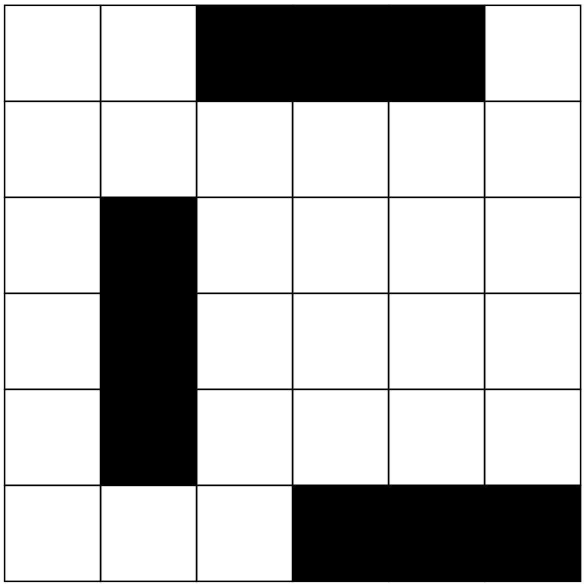


Abbildung 1: Beispiel eines von Schwarz gesetzten Spielfeldes

Zwei Spieler (Schwarz und Weiss) spielen das folgende Spiel: Schwarz setzt Schiffchen auf das Spielfeld (siehe Abbildung). Weiss "schiesst" auf freie Felder. Schwarz markiert jeden Schuß von Weiß auf dem Spielfeld. Ein Feld ist aus der Sicht von Weiss genau dann nicht frei, wenn Weiss bereits auf dieses Feld geschossen hat, wobei ein Schiffchen entweder getroffen oder nicht getroffen wurde. Weiss hat während des Spieles keine Kenntnis über die genaue Lage der von Schwarz markierten Schiffchen, wohl aber über die bereits beschossenen Felder.

Weiss gewinnt das Spiel, wenn Weiss alle Schiffchenpositionen nach maximal 18 Schüssen getroffen hat. Schwarz gewinnt das Spiel, wenn nach 18 Schüssen noch Schiffchenpositionen ungetroffen bleiben.

Vorgegeben ist eine Menge *Pos*, die das Spielfeld durch einen Mengenausdruck definiert. Vorgegeben ist ferner eine Klasse *Spiel* mit dem Zustandsschema und der Klasseninvarianten.

Spezifizieren Sie:

a.) das INIT-Schema von *Spiel* — 5 Punkte

b.) die Operationen — 35 Punkte
schwarz_setzt_schiffchen,
schwarz_markiert_getroffen,
schwarz_markiert_nicht_getroffen,
weiss_schiesst,
spiel_aus.

c.) Totalisieren Sie die Operation *zug_machen* — 10 Punkte
mit Hilfe der Operation *spiel_aus*
zur Operation *spiel_stand*.

Achten Sie darauf, dass *spiel_stand* einen geeigneten Ausgabeparameter *report!* hat.

d.) Welchen Typ hat *Pos* ? — 5 Punkte

Anleitung:

Beachten Sie, dass die Initialisierung der Zustandsvariablen, die die Schiffchenpositionen angibt, *nicht* im INIT-Schema spezifiziert wird; sie soll vielmehr von Schwarz vorgenommen werden. Deuten Sie das Setzen der Schiffchen auf dem Spielfeld lediglich durch Skizzieren eines geeigneten Mengenausdrucks an. Die genaue Angabe der Schiffchenpositionen soll unspezifiziert bleiben.

Schiffchen versenken Lösung der Teilaufgaben a) – c)

Spiel

$Pos == (1 \mathinner{..} 6) \times (1 \mathinner{..} 6)$

$Report ::= naechster\_zug \mid spiel\_ende$

$anzahl\_zuege : \mathbb{N}$
$getroffene\_felder : \mathbb{P} Pos$
$nicht\_getroffene\_felder : \mathbb{P} Pos$
$schiffchenpositionen : \mathbb{P} Pos$
Δ
$freie\_felder : \mathbb{P} Pos$

$getroffene\_felder \subseteq schiffchenpositionen$
$nicht\_getroffene\_felder \cap schiffchenpositionen = \varnothing$
$freie\_felder = Pos \setminus (nicht\_getroffene\_felder \cup getroffene\_felder)$
$nicht\_getroffene\_felder \cap getroffene\_felder = \varnothing$
$0 \leq anzahl\_zuege \leq 18$

Die Zustandskomponente *freie_felder* ist eine abgeleitete Größe: sie ist vollständig bestimmt durch die Komponenten *getroffene_felder* und *nicht_getroffene_felder* und darf nicht in einer Delta-Liste vorkommen.

INIT

$getroffene\_felder = \varnothing$
$nicht\_getroffene\_felder = \varnothing$
$anzahl\_zuege = 0$

Die Zustandkomponente *schiffchenpositionen* wird von Schwarz in *scharz_markiert_schiffchen* gesetzt und kommt daher in *INIT* nicht vor.

schwarz_setzt_schiffchen

$\Delta(schiffchenpositionen)$

$schiffchenpositionen = \varnothing$
$schiffchenpositionen' = \{(\ldots, \ldots), (\ldots, \ldots), (\ldots, \ldots),$
$(\ldots, \ldots), (\ldots, \ldots), (\ldots, \ldots), (\ldots, \ldots), (\ldots, \ldots), (\ldots, \ldots)\}$

Schwarz ist hier aktiv, daher ist der Effekt des Setzens der Schiffchenpositionen nicht von der Umgebung dieser Operation bestimmt, Die Operation hat daher weder Eingangs- noch Ausgangs-Parameter. Der Effekt wird allein von der Änderung einer Zustandgröße bestimmt. Wichtig ist hier die Vorbedingung: sie stellt sicher, dass das Feld nicht schon Schiffchen enthält. Eine vollständige Spezifikation dieser Operation folgt weiter unten.

$$
\begin{array}{|l}
\hline
\textit{schwarz\_markiert\_getroffen} \\
\Delta(getroffene\_felder) \\
pos? : Pos \\
\hline
pos? \in schiffchenpositionen \Rightarrow \\
\qquad getroffene\_felder' = getroffene\_felder \cup \{pos?\} \\
\hline
\end{array}
$$

Schwarz ist hier aktiv und erhält die zu markierende Position auf dem Spielfeld von der Umgebung (weiter unten zeigt sich: von Weiss!)

$$
\begin{array}{|l}
\hline
\textit{schwarz\_markiert\_nicht\_getroffen} \\
\Delta(nicht\_getroffene\_felder) \\
pos? : Pos \\
\hline
pos? \notin schiffchenpositionen \Rightarrow \\
\qquad nicht\_getroffene\_felder' = nicht\_getroffene\_felder \cup \{pos?\} \\
\hline
\end{array}
$$

$$
\begin{array}{|l}
\hline
\textit{weiss\_schiesst} \\
\Delta(anzahl\_zuege) \\
pos! : Pos \\
report! : Report \\
\hline
pos! \in freie\_felder \\
anzahl\_zuege < 18 \\
anzahl\_zuege' = anzahl\_zuege + 1 \\
report! = naechster\_zug \\
\hline
\end{array}
$$

Weiss ist hier aktiv; deshalb kann die zum Schuss ausgewählte Feldposition nicht von der Umgebung bestimmt sein. Wohl aber teilt Weiss der Umgebung (später zeigt sich, dass der Kommunikationspartner Schwarz ist!) die ausgewählte Position als Ausgabeparameter mit. Über die konkrete algorithmische Auswahl wird nichts ausgesagt, da ein solcher Algorithmus nicht spezifiziert werden sollte!!

$$
\begin{array}{|l}
\hline
\textit{spiel\_vorbei} \\
report! : Report \\
\hline
(\#getroffene\_felder = \#schiffchenpositionen) \lor anzahl\_zuege = 18 \\
report! = spiel\_vorbei \\
\hline
\end{array}
$$

$$
\begin{aligned}
zug\_machen \mathrel{\widehat{=}} weiss\_schiesst \| (&schwarz\_markiert\_getroffen \\
&\lor schwarz\_markiert\_nicht\_getroffen)
\end{aligned}
$$

Diese Operation spezifiziert die Kommunikation von Weiss zu Schwarz über den Kommunikationsparameter *pos*.

$$
spielstand \mathrel{\widehat{=}} zug\_machen \lor spiel\_vorbei
$$

Es folgt die genaue Spezifikation der Operation *schwarz_setzt_schiffchen*, die *nicht* Teil der geforderten Lösung ist.

$Spiel$

...

...

$schwarz\_setzt\_schiffchen$

$\Delta(schiffchenpositionen)$

$$
\begin{array}{l}
schiffchenpositionen = \varnothing \\
\exists (S1, S2, S3 : \mathbb{P}\,Pos \mid \\
\quad (\forall\, i, j : 1 \mathinner{..} 3;\; xi1, xi2, xi3, yi1, yi2, yi3 : 1 \mathinner{..} 6 \bullet \\
\quad\quad (\#Si = 3 \land \\
\quad\quad i \neq j \Rightarrow Si \cap Sj = \varnothing \land \\
\quad\quad Si = \{(xi1, yi1), (xi2, yi2), (xi3, yi3)\} \Rightarrow \\
\quad\quad\quad (((xi1 = xi2 = xi3) \land yi2 = yi1 + 1 \land yi3 = yi1 + 2) \\
\quad\quad\quad \lor \\
\quad\quad\quad (yi1 = yi2 = yi3) \land xi2 = xi1 + 1 \land xi3 = xi1 + 2))) \bullet \\
\quad schiffchenpositionen' = S1 \cup S2 \cup S3)
\end{array}
$$

...

Schwarz ist aktiv, daher werden die wichtigsten Eigenschaften der drei Schiffchen S1, S2, S3 spezifiziert:

- sie haben alle die Multiplizität 3,
- sie sind paarweise disjunkt (dürfen sich aber berühren),
- sie liegen in einer waagerechten oder senkrechten Linie,
- sie definieren durch ihre Vereinigung die Schiffchenpositionen.

...

d) Der Typ von Pos ist

$\mathbb{P}(\mathbb{Z} \times \mathbb{Z})$.

Aufgabe 3 (Statecharts)

(Geschätzte Bearbeitungszeit: 30 min.)

Auto fahren

Das Modell unseres Autos hat 4 parallel arbeitende Aggregate: *Zündung, Gangschaltung, Gas und Handbremse.*

Hinweis: Wir abstrahieren von der Bremse, d.h. sie soll nicht modelliert werden.

Im Schaubild (siehe nächste Seite) sind die Aggregate Gas und Handbremse bereits spezifiziert. Jedes Aggregat hat einen wohldefinierten Anfangszustand.

a.) Modellieren Sie *Gangschaltung*. *5 Punkte*

Die Gangschaltung unterscheidet die Zustände *LL (im Leerlauf)*, *R (im Rückwärtsgang)* und *V (in einem Vorwärtsgang)*. Es soll möglich sein, vom Leerlauf in den Rückwärtsgang oder in einen der Vorwärtsgänge und von dort zurück in den Leerlauf zu schalten.

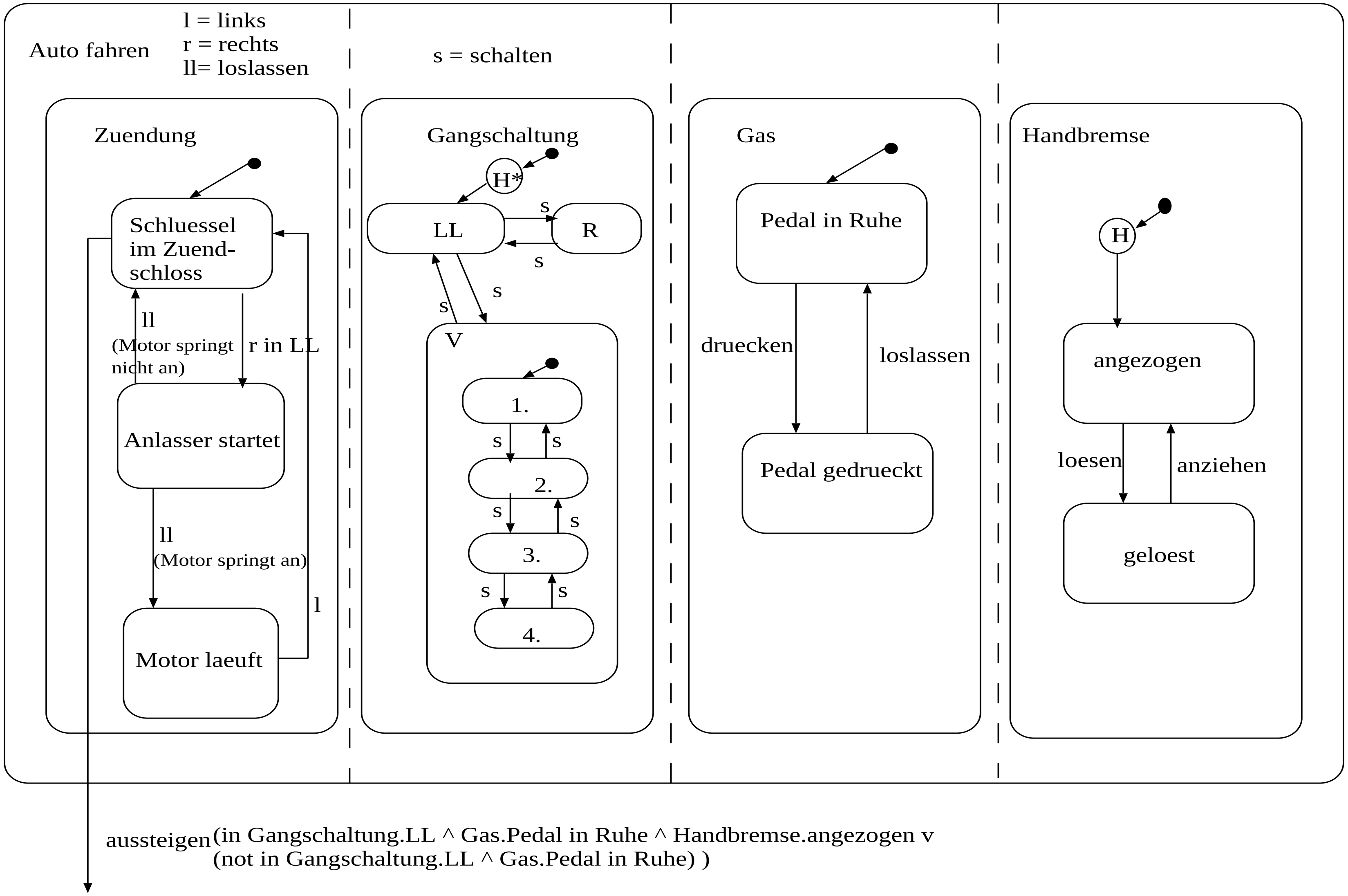
V soll weiter in Vorwärtsgänge verfeinert werden. Es soll möglich sein von jedem Vorwärtsgang in den nächsthöheren und in den nächstniedrigeren Gang zu schalten.

b.) Modellieren Sie *Zündung*. *10 Punkte*

Dabei sind die folgenden Zustände zu unterscheiden: *Schlüssel im Zündschloß*, *Anlasser startet*, *Motor läuft*. Achten Sie darauf, daß ein Startvorgang nur möglich ist, wenn kein Rückwärtsgang und keine Vorwärtsgänge eingelegt sind. Der Zustand *Motor läuft* wird nur unter der Bedingung erreicht, daß der Motor anspringt.

c.) Das Aussteigen aus dem Auto (das Ereignis *aussteigen* in Zündung) ist nur bei Kombinationen von bestimmten Zuständen aller Aggregate möglich. Spezifizieren Sie diese Bedingung. *15 Punkte*

Lösung



Auto fahren
l = links
r = rechts
ll= loslassen
s = schalten
Zuendung
Schluessel im Zuend-schloss
ll
(Motor springt nicht an)
r in LL
Anlasser startet
ll
(Motor springt an)
l
Motor laeuft
Gangschaltung
H*
LL
R
s
s
s
s
V
1.
2.
3.
4.
s
s
s
s
s
s
Gas
Pedal in Ruhe
druecken
loslassen
Pedal gedrueckt
Handbremse
H
angezogen
loesen
anziehen
geloest
aussteigen (in Gangschaltung.LL ^ Gas.Pedal in Ruhe ^ Handbremse.angezogen v
(not in Gangschaltung.LL ^ Gas.Pedal in Ruhe))