

# Blatt 5

## Software-Engineering II

### A 5.1: Zustandsdiagramm – Geldautomat

4 Punkte

Sie haben von einem Kollegen für einen Geldautomaten, dessen Funktionalität durch den folgenden Text beschrieben wird, eine Modellierung in Form eines UML Zustandsautomaten erhalten:

Der Geldautomat befindet sich zu Beginn in dem Zustand bereit. In diesem Zustand kann er Anfragen des Benutzer entgegennehmen. Der Benutzer initiiert seine Interaktion durch Eingabe seiner Bankkarte. Wenn die Karte ungültig ist, wird die Karte sofort ausgeworfen und der Geldautomat ist wieder bereit eine andere Karte anzunehmen. Ist die Karte gültig, wird der Benutzer aufgefordert, seine Geheimzahl einzugeben. Falls diese nicht gültig ist, ist eine erneute Eingabe der Geheimzahl möglich. Es sind hier maximal drei Wiederholung erlaubt. Nach erfolgreicher Eingabe der Geheimzahl ist es dem Benutzer möglich einen Betrag einzugeben, den er von seinem Konto abheben will. Ist dieser Betrag verfügbar (Deckung des Kontos gegeben, Geldautomat hat genug Scheine), wird das Konto mit dem entsprechenden Betrag belastet. Anschließend wird die Karte ausgegeben. Wird die Karte nicht aus dem Automaten entfernt, wird die Karte einbehalten und das Geld wird nicht ausgezahlt. In diesem Fall findet eine Rückbuchung des Betrags auf das Konto statt. Wird die Karte normal entnommen, wird der Geldbetrag dem Kunden zur Verfügung gestellt und anschließend kann ein anderer Kunde den Automaten verwenden.

#### UML Zustandsautomat

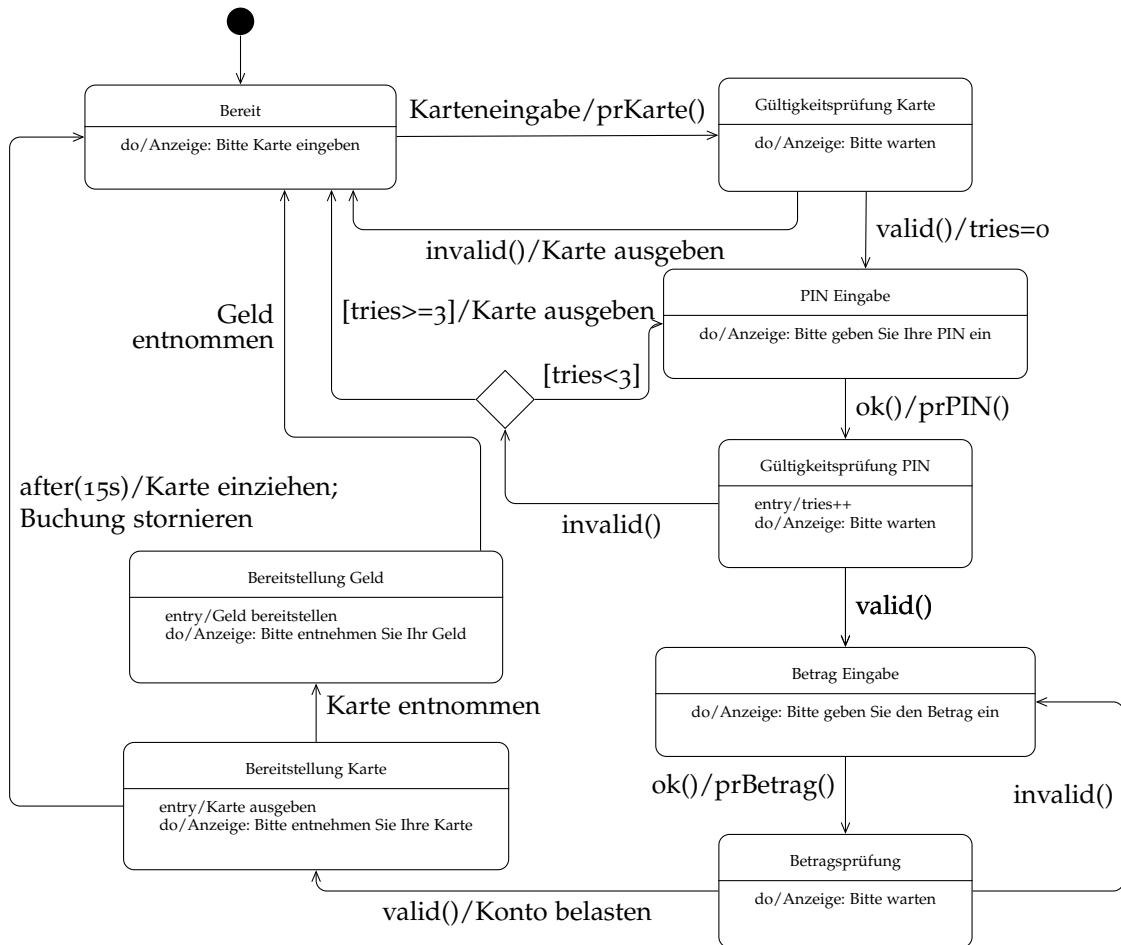
Es gibt die folgenden Ereignisse, auf die der Automat reagieren muss, je nach Zustand, in dem er sich gerade befindet:

- *Karteneingabe*: Ein Nutzer des Geldautomaten hat eine Karte in den Automaten eingeführt. Die Karte ist dann für den Nutzer nicht mehr entnehmbar.
- *valid()*: Eine Prüfung wurde durchgeführt, welche gültig war. Da die Prüfung selber nicht im Geldautomaten, sondern von einem Server durchgeführt wird, wird das Ergebnis der Prüfung in Form eines Call-Ereignisses modelliert.
- *invalid()*: Ein Prüfung wurde durchgeführt, welche ungültig war.
- *ok()*: Der Nutzer hat die Bestätigen Tasten auf dem Geldautomaten gedrückt um eine Eingabe abzuschließen.

- *Karte entnommen*: Der Nutzer hat die Karte aus dem Kartenschlitz entnommen.
- *Geld entnommen*: Der Nutzer hat das Geld aus dem Geldauswurf entnommen.
- *after(15s)*: Es sind 15 Sekunden vergangen und kein anderes Ereignis ist eingetreten. (Korrektur: in alter Version: *when(15s)*).

Es gibt die folgenden Aktivitäten, die der Automat initiieren kann:

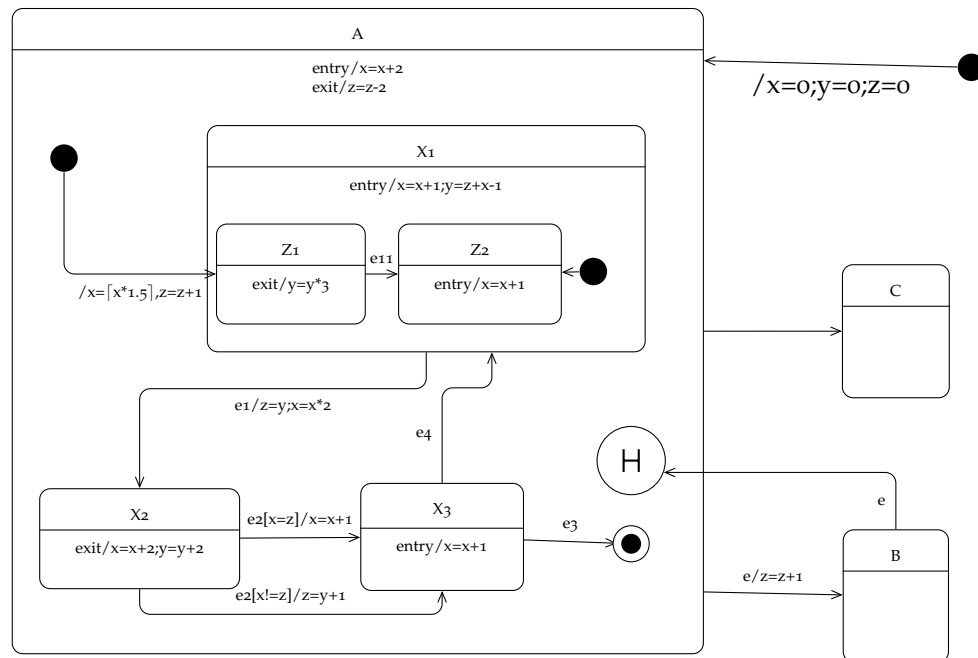
- *prKarte()*, *prPin()*, *prBetrag()*: Prüfe den entsprechenden Wert. Bei der Karte wird z.B. die Kartenummer ausgelesen und an eines Server geschickt. Dieser stellt sicher, dass die Kartenummer nicht auf der Liste der gesperrten Karten steht. Ist die Karte komplett nicht lesbar, führt die Prüfung ebenfalls zum Ablehnen der Karte.
- *Geld bereitstellen*: Das Geld wird dem Nutzer zur Verfügung gestellt, so dass dieser das Geld entnehmen kann.
- *Karte ausgeben*: Die Karte wird dem Nutzer bereitgestellt. Dieser kann die Karte wieder entnehmen.
- *Karte einziehen*: Die Karte wird innerhalb des Geldautomaten in einer Sammelbox aufbewahrt. Der Einfachheit halber wird angenommen, dass die Sammelbox beliebig viele Karten aufnehmen kann.
- *Konto belasten*: Das Konto des Kartenbesitzers wird mit dem eingegebenen Betrag belastet.
- *Buchung stornieren*: Die durchgeführte Buchung wird storniert. Es wird kein Geld von dem Konto des Kartenbesitzers abgebucht.



Nach kurzem Betrachten der Beschreibung und des Automaten stellen Sie fest, dass eine wichtige Funktionalität übersehen wurde. Der Nutzer kann das Abheben von Geld nicht abbrechen. Erweitern Sie den vorhandenen Zustandsautomaten so, dass dieser Vorgang im Sinne des Nutzers eingebaut wird. Dokumentieren Sie, wie dies oben der Fall ist, welche zusätzlichen Ereignisse auftreten, welche zusätzlichen Aktivitäten entstehen, und wie der Automat selber anzupassen ist.

## A 5.2: Zustandsdiagramm – Ablauf

6 Punkte



Simulieren Sie auf dem oben angegebenen Zustandsdiagramm den Ablauf der folgenden Ereignisse:

- START, e1, e2, e, e, e2, e4
- START, e, e2, e, e11, e1

Führen Sie die zweite Ereignisfolge auf einem leicht modifizierten Zustandsautomaten aus, indem Sie den History Zustand durch einen tiefen History Zustand ersetzen.