

Hausaufgabe 4

Tim Wende

28. Oktober 2021

1 Aktivitätsdiagramm Fahrkartenautomat

In [Übung 3](#) wurde ein Lastenheft für einen Fahrkartenautomaten erstellt.

- a. Dokumentieren Sie den Anwendungsfall „Fahrkarte kaufen“ mit Hilfe eines **UML-Aktivitätsdiagrammes**, das z.B. mit [Visual Paradigm](#) erstellt wird.

Teilen Sie das Aktivitätsdiagramm dabei in die folgenden Partitionen (Schwimmbahnen) auf:

- Kunde: Interaktion des Kunden mit dem Automaten
- Display: Anzeige von Text
- Automat: Eingaben einlesen, Rückgeldberechnung, Kartendruck, ...

Modellieren Sie dabei Geld und Fahrkarten als **Objektknoten**! Hinweise:

- Der Automat arbeitet nur mit Bargeld.
- Beachten Sie, dass der Bezahlvorgang auch abgebrochen werden kann.
- Verwenden Sie Parallelitäten, wo es für Sie Sinn ergibt.
- Beachten Sie auch den Fall, wenn der Automat nicht genug Rückgeld geben kann.

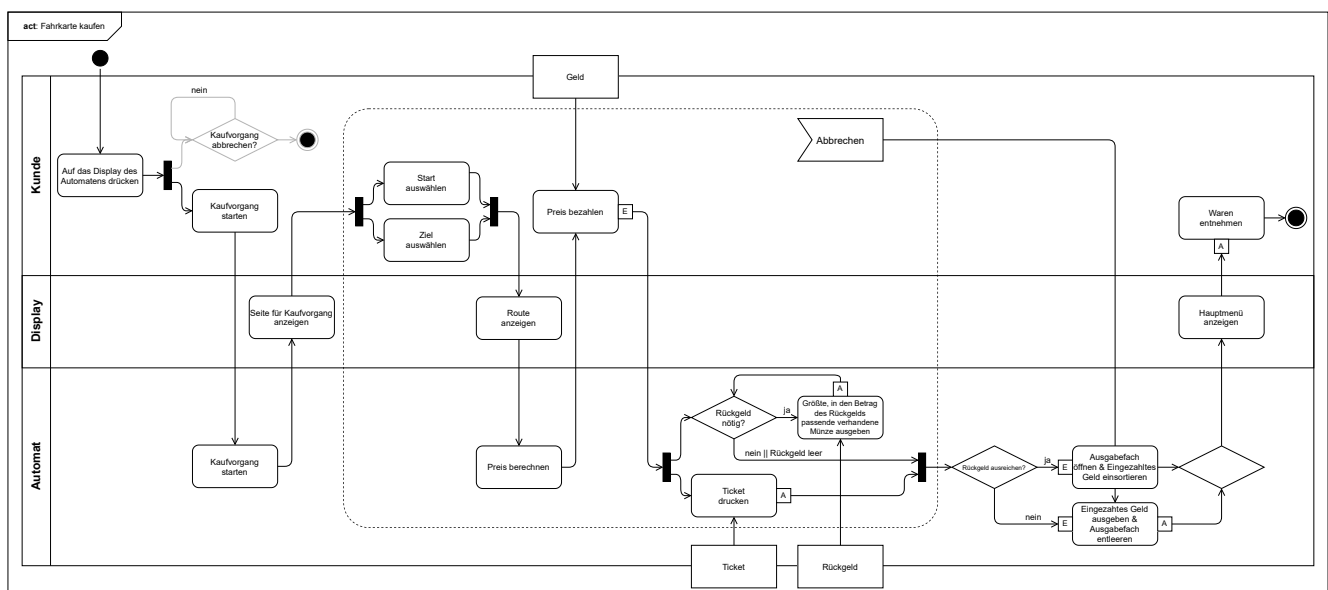


Abbildung 1: activity_diagram

Kommentare zum Diagramm:

- Die „Kaufvorgang abbrechen“-Loop in der oberen linken Ecke ist parallel möglich, jedoch ist ein Interrupt zugegebenermaßen sinnvoller. Allerdings war die *if*-Abfrage bereits fest in meinem Design verankert, also hier als alternative Darstellung ausgegraut angefügt.
- Um in dem Diagramm verschiedene **Objektknoten** gleichzeitig mitgeben zu können, habe ich den jeweiligen Paketen verschiedene Tags verliehen:
 - E: stellt hier alle Eingaben des Benutzers dar
 - A: stellt alle Ausgaben des Automaten dar

Ich bin mir nicht sicher, ob das von UML 2.0 so vorgesehen ist, jedoch empfand ich es als sinnvoll.

- Mein Automat besitzt ein eigenes Ausgabefach, welches nach Belieben geöffnet und geschlossen werden kann. Somit ist es für den Automaten möglich, quasi jede Münze des Rückgelds einzeln auszugeben, und im späteren Verlauf alle wieder einzusammeln, ohne dass der Kunde etwaige entnehmen kann. Da sich alle physischen Ausgaben des Automaten bis zum absolut sicheren Ende des Kaufvorgangs in diesem Fach befinden, können „Ticket drucken“ und „Rückgeld ausgeben“ parallel ablaufen.
- Um jedoch keine unnötigen Tickets zu drucken, ist es in meinem Automaten nicht möglich, dass der Münzspeicher leer wird. Wie man in meiner Abgabe zu Woche 3 erkennen kann, wird bei niedrigem Füllstand der Münzen ein Event ausgelöst, sodass ein Mitarbeitender der Administration diesen auffüllen kann. Dies ist zwar ökologisch, jedoch nicht realitätsnah (aber wir sind ja auch in SWT; ist also normal).
- Entschuldigung für die kleine Schrift, jedoch handelt es sich bei der eingefügten Grafik um eine .pdf-Datei. Fühlen Sie sich also frei nach Belieben reinzuzoomen.

- b. Beschreiben Sie mit eigenen Worten, wie der Zusammenhang zwischen UML-Use-Case-Diagrammen, speziell der Szenariobeschreibung, und den UML-Aktivitätsdiagrammen ist.

Ein UML-Aktivitätsdiagramm beschreibt einen Pfad des Anwendungsfalldiagramms. Somit wird sich aus der Sicht eines beliebigen Stakeholders ein Anwendungsfall angeschaut und bis zum jeweiligen Ende weitergeführt. Wenn man sich die Szenariobeschreibung genauer anschaut, kann man mit cleverem Kombinieren des typischen sowie aller alternativen Abläufe ein Aktivitätsdiagramm erstellen. So spiegeln sich alle möglichen Ausgangszustände sowohl im Anwendungsfalldiagramm als auch im Aktivitätsdiagramm wider. Des Weiteren findet man die benötigten Objekte in den Partitionen (je Stakeholder) und in den Ein- sowie Ausgabeflächen als **Objektknoten** wieder.

Schaut man sich beispielsweise den Anwendungsfall „Fahrkarte kaufen“ an, sieht man im Aktivitätsdiagramm relativ schnell den typischen Ablauf:

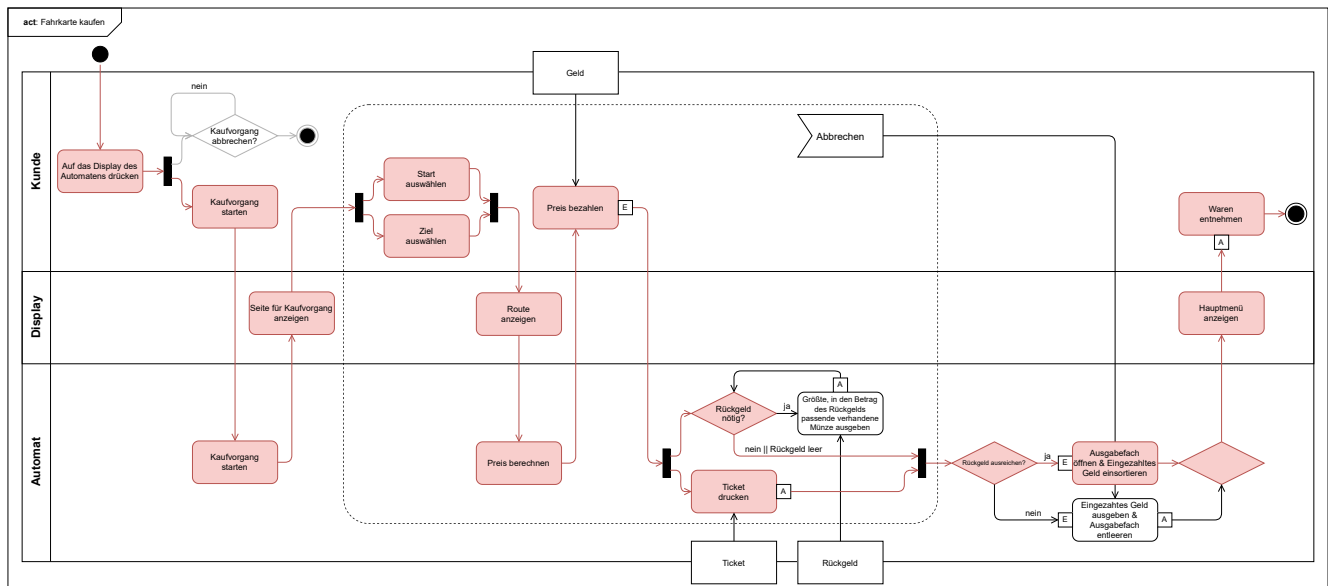


Abbildung 2: activity_diagram_typical

Alternativ habe ich hier mal einen abzubrechenden Ablauf dargestellt. Der Abbruch-Zeitpunkt ist offensichtlich variabel, ließe sich also innerhalb der **interruptable region** beliebig verschieben

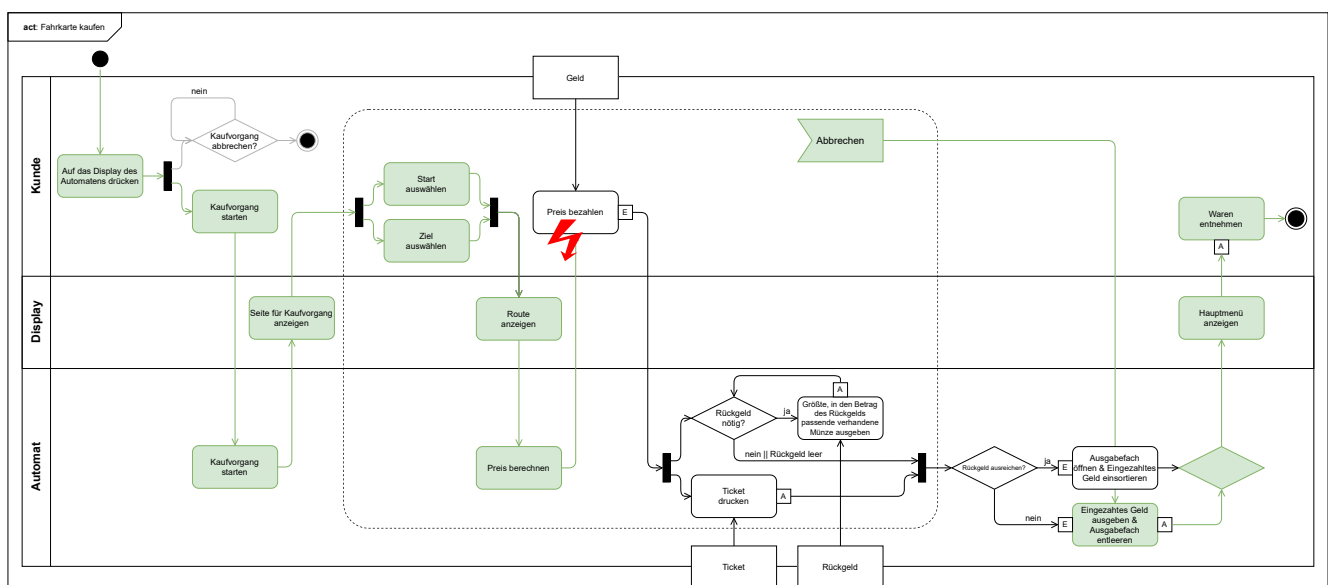


Abbildung 3: activity_diagram_abort

2 Klassendiagramm & Objektdiagramm

Aus einer Protokoll-Notiz sind folgende Informationen festgehalten:

Ein **Reisender** kann eine **Einzelfahrkarte** zu einem bestimmten Tarif (als Zeichenkette) und einen Preis (EUR) erwerben. Diese Fahrkarte *legitimiert* den Reisenden zur *Nutzung* einer bestimmten **Strecke**, die zwei **Stationen** miteinander *verbindet*. Die Strecke *kennt* dabei das Ausgangs- und Zielgleis. Die Station ist über den Stationsnamen eindeutig gekennzeichnet, der Reisende über seinen Namen.

- a. Erstellen Sie ein fachliches Klassendiagramm. Identifizieren Sie hierzu die Klassen, deren Beziehungen und die Attribute mit Datentypen. Modellieren Sie noch keine Methoden (Anm. d.Verf.: „:“) Achten Sie auf die Multiplizitäten und Assoziationen. Alle Assoziationen sollen einen Namen und eine Leserichtung aufweisen.

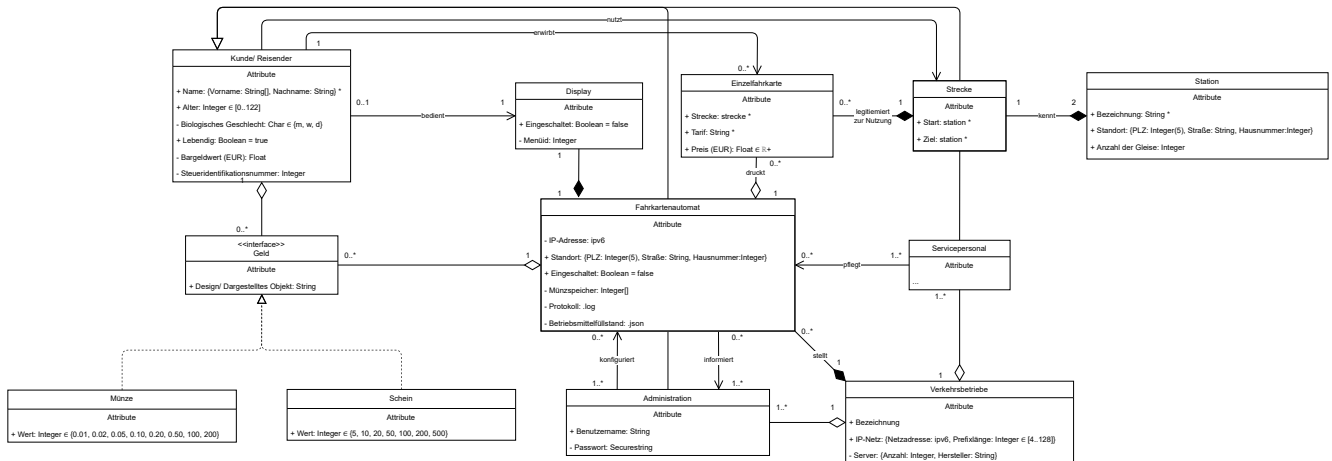


Abbildung 4: class_diagram

* Aus der Aufgabenstellung bekannt (Rest aus h03 übernommen)

- b. Erstellen Sie auch ein passendes Objektdiagramm, das folgenden Sachverhalt beschreibt:
Die Kundin Frau Müller fährt für 17.50€ mit dem Gute-Nacht-Tarif von Aachen (Gleis 7) nach Köln Hbf (Gleis 8)
Hinweis: Sie brauchen im Objektdiagramm keine Datentypen, Multiplizitäten, Leserichtungen und Assoziationsnamen zu modellieren.

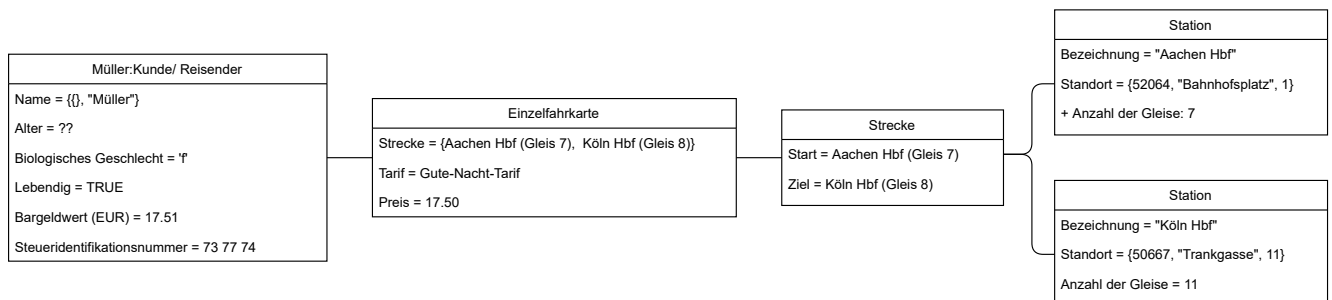


Abbildung 5: object_diagram