

Übungsblatt 3

Lösungsvorschlag
29.06.2014

Aufgabe 1)

Der Anwendungsfall dieser Aufgabe lässt sich am einfachsten anhand des zentralen Prozesses *Einschreibung an der Universität* schlicht nach diesem benennen.

In Bezug auf diesen Anwendungsfall gibt es drei interagierende Akteure zu benennen, wobei mit StudentIn und SachbearbeiterIn die beiden grundsätzlich am Prozess beteiligten Akteure gelistet sind und es als eine spezifische Untergruppe von StudentIn auch noch die Variation InternationaleR StudentIn gibt.

Der Prozess selbst wird unter der Bedingung eingeleitet, dass vonseiten eines Studenten Interesse an der Einschreibung an der Universität besteht, sowie dass eine Sachbearbeiterin zur Bearbeitung der Aufgabe zur Verfügung steht und auf sämtliche notwendigen Systeme und Daten zugegriffen werden kann.

Beim eigentlichen Prozessablauf tätigen Mitarbeiter und Student die Einschreibung dergestalt gemeinsam, als dass die für die Einschreibung nötigen Informationen vom Studenten an die Mitarbeiterin ausgegeben werden, die die Einschreibung vornimmt, zuletzt kann auch eine Einschreibung für einen Studiengang durchgeführt werden, auch wenn dies nicht notwendigerweise Teil des Prozessablaufs sein muss. Die Einschreibung für ein Studium Generale steht in einer Vererbungsbeziehung zur Einschreibung an der Universität und in dieser Hinsicht in Beziehung zu diesem Prozessablauf.

Die nennenswerte Variation im Prozessablauf basiert auf der Vorbedingung dass der Akteur StudentIn der Gruppe InternationaleR StudentIn zugeordnet ist, in diesem Falle kann von jener Seite aus zuvor eine Stellungnahme des International Office als zusätzlich für den Prozess anwendbarer Datensatz eingeholt werden.

Aufgabe 2)

Zur Auswertung der in der Aufgabe gelisteten Szenarien bietet sich ein Vorgehen nach dem Schema der klassischen textuellen Beschreibung von Anwendungsfällen an, wobei einleitend gesagt sei dass die in den Szenarien gelisteten Nutzfälle der Informationssuche durch Privatkunden und Großkunden, sowie der Händlerbestellung und Direktbestellung entsprechen, wobei als beteiligte Akteure Privatkunde, Großkunde, Händler, Disponent zu listen sind.

Szenario 1

Name:

Informationszugriff und Konfiguration - Privatkunde

Akteure:

Privatkunde

Vorbedingung:

Privatkunde wünscht Informationen zu spezifischem Produkt

Privatkunde hat Zugang zum Informationssystem

Informationssystem ist einsetzbar/verfügbar

Nachbedingung:

Privatkunde hat eigene Produktkonfiguration erstellt

Ablauf:

Privatkunde greift auf das Informationssystem zu

Privatkunde sucht nach dem Modell X123

Informationssystem zeigt Kenndaten von Modell X123 an

Privatkunde konfiguriert Kenndaten im Sinne eines Wunschmodells

Szenario 2

Name:

Informationszugriff - Großkunde

Akteure:

Großkunde

Vorbedingung:

Großkunde wünscht Informationen über spezifisches Modell

Großkunde hat Zugang zum Informationssystem

Informationssystem ist einsetzbar/verfügbar

Nachbedingung:

Keine Nachbedingung

Ablauf:

Großkunde greift auf das Informationssystem zu

Großkunde sucht nach dem Modell X987

Informationssystem zeigt Kenndaten von Modell X987 an

Großkunde sucht nach dem Modell X567

Informationssystem zeigt Kenndaten von Modell X567 an

Großkunde beendet das Informationssystem

Szenario 3

Name:

Händlerbestellung - Privatkunde

Akteure:

Privatkunde

Händler

Disponent

Vorbedingung:

Privatkunde möchte ein Modell bestellen

Händler ist verfügbar

Disponent ist verfügbar

Informationssystem/Bestellsystem ist verfügbar

Produktionssystem ist verfügbar

Nachbedingung:

Bestellung ist abgeschlossen

Produktionsauftrag ist erstellt

Ablauf:

Privatkunde sucht Händler auf

Händler erhält Bestelldaten von Privatkunde

Händler ruft Informationssystem/Bestellsystem auf

Händler sucht nach Modell X234
Informationssystem/Bestellsystem zeigt Datenblatt zu Modell X234 an
Händler konfiguriert das Modell nach den Wunschdaten des Kunden
Händler nutzt integrierte Bestellfunktion zur Bestellung von Modell X234
Informationssystem/Bestellsystem leitet Bestellauftrag an Disponenten weiter
Disponent bestätigt Bestellung und schließt damit die Bestellung ab
Händler beendet Informationssystem/Bestellsystem
Disponent ruft Produktionssystem auf
Disponent erstellt Produktionsauftrag
Disponent beendet Produktionssystem

Szenario 4

Name:

Direktbestellung - Großkunde

Akteure:

Großkunde

Händler

Disponent

Vorbedingung:

Großkunde möchte ein Modell bestellen

Alle haben Zugriff auf das Informationssystem/Bestellsystem

Informationssystem/Bestellsystem ist verfügbar

Disponent hat Zugriff auf Produktionssystem

Produktionssystem ist verfügbar

Nachbedingung:

Bestellung ist abgeschlossen

Produktionsauftrag ist erstellt

Ablauf:

Großkunde ruft Informationssystem/Bestellsystem auf

Großkunde sucht nach Wunschmodell

System zeigt Datenblatt an

Großkunde konfiguriert die Modelldaten nach Wunsch

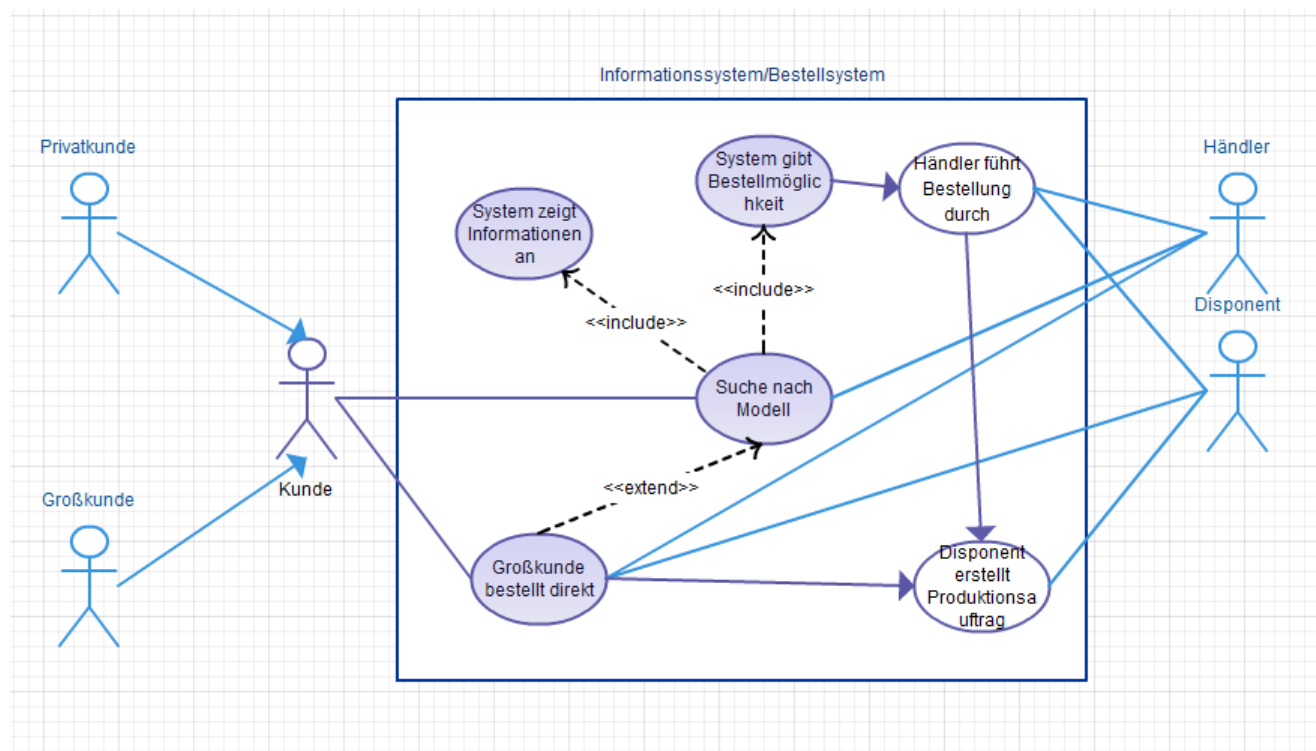
Großkunde gibt Bestellung auf

Händler und Disponent bestätigen Bestellung

Disponent ruft Produktionssystem auf

Disponent gibt Produktionsauftrag aus

Alle Szenarien lassen sich einem Anwendungsfalldiagramm darstellen um den Prozessablauf schematisch zu überblicken und zu verdeutlichen.



Aufgabe 3)

Assoziation:

Assoziationen sind eine Relation, welche in UML Klassendiagrammen angewandt wird um eine Beziehung zwischen mehreren Objektinstanzen anzuzeigen. Besondere Bedeutung hat hierbei der Umstand, dass diese Assoziationen auch eine Navigierbarkeit implizieren können, was gestalterisch durch Pfeile anzeigbar ist.

Diese Navigierbarkeit ist notwendig um im Rahmen der Implementierung erkennen zu können ob irgendwo ein Wechsel vom Einem Objekt zum Anderen realisiert werden muss und ist zum Beispiel durch die Speicherung einer Objektreferenz in einem Attribut umsetzbar.

Aggregation:

Die Aggregation kann als speziellere Form der Assoziation verstanden werden, welche dazu dient eine vereinigende Beziehung von einzelnen Objekten in einem Aggregat zu symbolisieren. Durch diese Maßnahme lassen sich also übergeordnete Aggregate aus unterschiedlich vielen Objekten bilden, um so eine bessere Ordnung oder eine genauere Spezifikation zu ermöglichen.

Eine beispielhafte Möglichkeit hierfür wäre die Unterordnung verschiedener Bücher in eine Buchreihe.

Komposition:

Die Komposition lässt sich leicht als erweiterte und stärkere Form der Aggregation verstehen, wobei hier signalisiert wird, dass das Aggregat und seine Komponenten logisch derartig miteinander verbunden sind, als dass die Komponenten selbst nicht ohne das Aggregat vorhanden sein können, es existiert also eine ständige und starke Bindung zwischen beiden, diese beruht auch darauf, dass eine Komposition nur zwischen den Komponenten und jeweils einem Aggregat bestehen kann.

Als logisches Beispiel könnte man eine Buchreihe durch Komposition mit einem Verlag darstellen, um zu signalisieren dass eine Buchreihe nicht ohne den dazugehörigen Verlag existieren

kann, auch wenn diese Beispiel im Zusammenhang mit der modernen Eigenpulikation vielleicht an Schärfe verlieren kann.

Generalisierung:

Generalisierungen sind eine praktische Relation zur Umsetzung von Vererbungshierarchien auf der schematischen Ebene von Oberen und Unteren Klassen, sie kennzeichnen voneinander ererbende Klassen und geben gleichzeitig die Richtung in der Erbhierarchie an, wobei die Unteren Klassen die Attribute und Methoden der Oberen Klassen erben, selbige aber auch beliebig umschreiben und erweitern oder aber auch neue Methoden und Attribute definieren können. Im Rahmen der Generalisierungen können auch abstrakte Klassen angelegt werden, die nur zur Einfachheit mehrere Eigenschaften vererbbar in sich vereinigen, als Objekt angewandt jedoch nicht vorkommen.

Als beispielhafte Umsetzung könnte man eine Klasse *Person* als abstrakte Oberklasse innerhalb des eigenen Diagramms einsetzen um hierin alle generellen Eigenschaften der in der Darstellung auftauchenden Akteure zu vereinen, sodass man für die tatsächlich beteiligten Akteure nur Rücksicht auf gesonderte Eigenschaften nehmen muss, da alle allgemeinen Eigenschaften bereits über die Vererbungshierarchie aufgeführt und in Beziehung gesetzt wurden. Damit wären Klassen wie *Ausleiher* oder *Bibliothekar* lediglich Subklassen von *Person* mit wenigen zu listenden Sondereigenschaften.

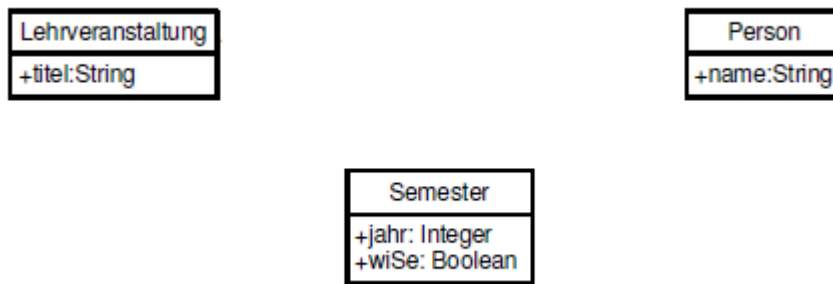
Aufgabe 4)

Aufgabe 5)

a)

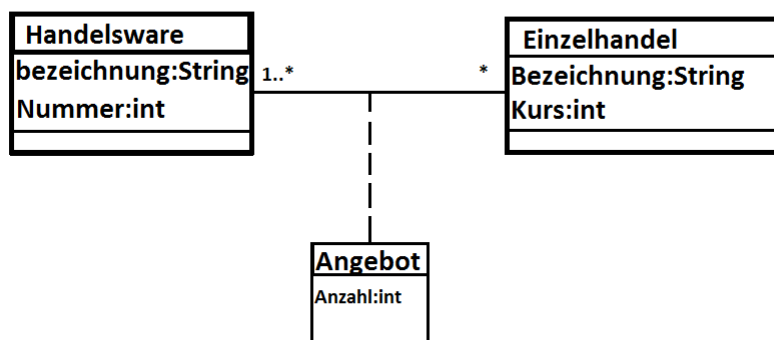
Die beiden Diagramme modellieren eine Lehrveranstaltung im Bezug auf die Beziehung zwischen den an der Veranstaltung teilnehmenden Personen und der Lehrveranstaltung, welche beide als Klassen realisiert sind und zwischen jenen sowie dem ebenfalls als Klasse umgesetzten Semester, in welchem die Teilnahme stattfindet. In allen Diagrammen sind dabei dieselben Klassen umgesetzt, so existieren gleichermaßen die Klassen *Person* und *Lehrveranstaltung*, die jeweils die Eigenschaften *name* und *titel* vom Typ String aufweisen, wie auch die Klasse *Semester*, welche die Eigenschaften *jahr* vom Typ Integer, sowie *wiSe* vom Typ Boolean besitzt. Der vornehmlichste Unterschied zwischen den Diagrammen besteht also nicht in den existierenden Klassen selbst, sondern in der Modellierung ihrer Beziehungen untereinander, hier besteht die Gemeinsamkeit nur in einer Assoziation *nimmt teil an*, die von *Person* nach *Lehrveranstaltung* navigierbar ist.

Im Rahmen eines Objektdiagramms lassen sich die zuvor besprochenen Klassenobjekte noch einmal verdeutlichen.



Beim ersten Diagramm besteht diese Beziehung weiterhin auch als einzige einfache Assoziation in der Modellierung, die Klasse *Semester* ist hierbei nämlich als Assoziationsklasse an die Beziehung *nimmt teil an* angegliedert, weitere Assoziationen bestehen in der Darstellung nicht. Diese Form der Modellierung einer mehrstelligen Assoziation ist dann praktisch, wenn gewünscht wird eine Beziehung mit Attributen zu versehen und damit eine Beziehung zu beschreiben, welche gleichzeitig die Funktion einer Klasse einnimmt, um so die Klassen *Person* und *Lehrveranstaltung* in eine spezifischere Beziehung zu setzen.

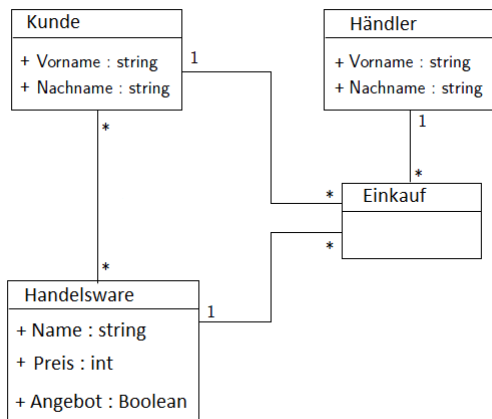
Ein Beispiel für diese Umsetzung ließe sich folgendermaßen gestalten,



Somit werden in dieser Modellierung die Attribute von *Semester* direkt der Assoziation *nimmt teil an* zugeordnet, was einerseits die Übersichtlichkeit in der Darstellung erhöht, vor allem aber das Abrufen der Attribute im Rahmen der Assoziation ermöglicht, statt eine neue Assoziation zur Klasse *Semester* anlegen und abrufen zu müssen.

Das zweite Diagramm gestaltet sich in dieser Hinsicht anders, da es auf eine Assoziationsklasse verzichtet und stattdessen die Klasse *Semester* als separate Klasse realisiert, welche in ihre Richtung navigierbare Assoziationen zwischen ihr und der Klasse *Lehrveranstaltung*, in Form der Assoziation *findet statt in*, und der Klasse *Person*, über die Assoziation *nimmt teil in* aufweist.

Diese Darstellung bettet die Attribute der Klasse *Semester* nicht in den Rahmen einer Assoziation ein, sondern verknüpft sie mit den beiden anderen Klassen über jeweils eine gesonderte Assoziation, eine Modellierung, welche sich im Ausbau auch als Realisierung einer mehrstelligen Assoziation durch den Einsatz einer verbindenden Klassen anwenden lässt. Dieser eher klassische Ansatz erlaubt es dann nach folgendem Beispiel mehrere Klassen zugleich miteinander zu assoziieren.



Konkret unterscheidet sich diese Darstellungsform dahingehend, als dass die Attribute von *Semester* eben nicht nur einer Assoziation zugeordnet werden, sondern über separate Beziehungen abgerufen werden können, was je nach Interessenslage gefragt sein kann.

b)

Gehen wir von der Annahme aus, dass die Attributierung von *Semester* im Rahmen der Assoziation *nimmt teil an* gefragt ist, so würde ich mich definitiv für die erste Modellierungsvariante entscheiden, da diese eine klare Zuordnung jener Klasse zur Beziehung selbst kennzeichnet. Auf diese Weise ist in übersichtlicher Weise dargestellt dass im Rahmen der eigentlichen Beziehung der Teilnahme bereits ein Mechanismus zum Abrufen der Attribute zu implementieren ist.

Da auch eine Assoziationsklasse noch mit weiteren Klassen assoziiert werden kann ist hier auch noch eine weitere Verknüpfung möglich, falls der gesamte Fall ausgebaut werden muss, auch wenn hier die Realisierung als mehrstellige Assoziation über eine Klasse übersichtlicher sein könnte.

Aufgabe 6)

Das vorliegende Klassendiagramm weist mehrere Mängel auf, so sind die Multiplizitäten der einzelnen Assoziationen nicht angegeben wodurch die Standardmultiplizität 1 bei einer Verbindung zu vielen Objekten wie bei der Klasse *Gruppe* irreführend, ebenso sind die Assoziationen nicht benannt und weisen keine definierten Rollen auf, auch ließe sich die Aggregation zwischen *TutorIn* und *Lehrkraft* anders modellieren und Irreführung zu vermeiden und zuletzt sollten Attribute in der Regel versteckt werden, statt in Klassendiagrammen zu Entwurf oder Implementierung öffentlich sichtbar zu sein.

Aufgabe 7)