

5. Übungsblatt

Ausgabe: 03.03.06

Abgabe: 31.03.06

Nach dem spektakulären Zusammenbruch vor ein paar Jahren fängt sich auch der Markt für Internetfirmen wieder, und das Stichwort Dott-Komm löst nicht mehr grundsätzlich nur Heiterkeit aus (“easy dot com, easy dot gone”). Immer mit dem Auge auf ein paar zusätzliche Einkommensquellen wollen wir diese Gelegenheit nutzen. Leider verfügen wir weder über die erforderlichen Mittel für eine umfangreiche Marktstudie noch über die kreativen Ideen um eine aussichtsreiche Marktlücke zu finden, also kopieren wir einfach eine Idee, die sich schon mehrfach bewährt hat — warum sollte sie nicht für uns auch funktionieren: wir verkaufen Bücher über das Netz.

10 *Mein kleiner Buchladen*

30 Punkte

Ein Buch in unserem Sinne hat einen Autor, ein Titel, ein Sachgebiet, eine (eindeutige) ISBN-Nummer und eine (im Rahmen des Sortiments) eindeutige Artikel-Nummer. Als Sachgebiete kennen wir zum Beispiel Sachbücher, Belletristik, Kinderbücher und Ratgeber.

Unser Web-Shop besteht aus drei Teilen: dem *Katalog*, in dem eine Reihe von Büchern verzeichnet sind, dem *Lager*, welches aus einer Menge von vorrätigen Büchern besteht, und dem *Web Front-End*, welches den Kunden erlaubt, den Katalog zu durchsuchen und Bücher zu bestellen. Letzteres ist der Schwerpunkt der hier zu erstellenden Modellierung.

Die Hauptseite (“Portal”) des Web-Frontends sollte folgende Möglichkeiten bieten:

1. Sich als neuer Benutzer *registrieren*: Dazu gibt der Benutzer seine Daten ein (Vorname, Nachname, Lieferadresse, Kontoverbindung oder Kreditkartennummer, sowie eine frei gewählte Benutzerkennung und ein Paßwort). Es wird geprüft, ob die Benutzerkennung noch nicht vergeben ist.
2. Sich als registrierter Benutzer *anmelden*: Dazu gibt der Benutzer seine Benutzerkennung sowie das Passwort ein. Wenn das Passwort mit dem bei der Anmeldung eingegeben übereinstimmen, ist der Benutzer unter dieser Kennung angemeldet. Nicht angemeldete Benutzer sind “Gäste”.
3. Den Katalog zu *durchsuchen*: der Benutzer kann Autor, Titel, ISBN-Nummer oder Sachgebiet (oder Kombinationen davon) angeben, und es werden alle Bücher angezeigt, die diesen Kriterien entsprechen und im Lager vorrätig sind. Ein angemeldeter Benutzer kann angezeigte Bücher dann *kaufen*, in dem er sie in seinen *Warenkorb* legt; der Warenkorb ist einem jedem angemeldeten Benutzer zugeordnete Menge von Büchern, die dieser zu kaufen beabsichtigt.
4. *Bezahlen*: Dazu wird der Inhalt des Warenkorbs sowie der Gesamtpreis angezeigt. Der Benutzer muss diese Bestellung bestätigen, danach wird sie versandt, und das Konto des Benutzers mit dem Warenpreis belastet.

Der Versand wird wie folgt modelliert: wir nehmen an, es gibt eine Versandabteilung. Diese akzeptiert Versandorder, die aus einer Adresse sowie einer Liste von Artikelnummern bestehen. Diese Order können durch das Setzen einer geeigneten Ausgangsvariablen (z.B. *!Versandorder*) abgeschickt werden. Die Lagerhaltung muss noch gesondert betrieben werden (d.h. versandte Objekte müssen noch aus dem Lager ausgetragen werden).

5. Sich *abmelden* (dabei wird der Warenkorb gelöscht).
6. Ein Objekt wieder aus dem Warenkorb *herausnehmen*.

Hinweise:

1. Modellieren Sie zuerst den Gesamtzustand des Systems: er besteht zum einen aus dem Katalog, dem Warenbestand, den Benutzerdaten und den Daten des aktuellen Benutzers (d.h. der Benutzerkennung und ggf. dem Inhalt seines Warenkorbs) . Dabei können Sie stark vereinfachend davon ausgehen, dass immer nur eine Benutzer zur Zeit das System benutzt (um uns Nebenläufigkeitsprobleme zur Zeit zu ersparen).
2. Zur Modellierung der Operationen kann es sinnvoll sein, einen Zustandsgraphen zu erstellen, dessen Knoten die verschiedenen Systemzustände sind, und dessen Kanten Zustandsübergängen entsprechen. Jeder Kante (jedem Systemzustandsübergang) entspricht (mindestens) einem Schema. Die Kanten können mit Bedingungen versehen sein (z.B. neue Benutzerkennung eindeutig).
3. Danach modellieren Sie jede Operation oben mit einer Schemaoperation (die natürlich noch verschiedene Hilfsoperationen benötigen können). Die Auswahl des Benutzers erfolgt durch Eingabevariablen. Zum Beispiel modellieren wir für die Hauptseite die möglichen Aktionen wie folgt:

Actions ::= Register | Login | Browse | Pay | Logout | RemoveFromBasket

Jedes der einzelnen Schema für registrieren, einloggen, etc. hat dann die Bedingung, dass die aktuelle Benutzereingabe der jeweiligen Aktion entspricht, zum Beispiel:

<i>Register</i> <i>?UserInput : Actions</i> ...	<i>?UserInput = Register</i> ...
---	---

Dieses Schema müßte jetzt die neuen Benutzerdaten, gelesen aus entsprechenden Eingabevariablen, den Benutzerdaten des Systems hinzufügen. Dabei muß geprüft werden, ob die neue Benutzerkennung eindeutig ist; es wird also zwei Zustandsübergänge geben (eine, wenn diese Bedingung erfüllt ist, und eine im Fehlerfall).

4. Wichtig ist, dass die Modellierung vollständig ist, d.h. jeder mögliche Zustandsübergang ist definiert (wichtig bei Bedingungen wie z.B. der Eindeutigkeit der Benutzerkennung; siehe auch das Texteditor-Beispiel aus der Vorlesung). Skizzieren Sie einen

Beweis der Vollständigkeit (dazu müssen Sie zeigen, dass es für jede mögliche Eingabe immer einen Zustandsübergang gibt; wenn es Vorbedingungen gibt, muss die Disjunktion dieser Vorbedingungen wahr sein).

5. Bitte die Modellierung ausführlich *kommentieren*! Zur Formatierung empfehlen wir L^AT_EX und das Paket `oz.sty` (siehe Webseite).
6. Die obigen Modellierung ist natürlich unvollständig, es fehlt die Möglichkeit, den Katalog und den Lagerbestand zu verwalten. Das ist so gewollt (damit die Aufgabenstellung nicht beliebig ausufert).

11 *Data Warehousing*

10 Punkte

Erweitern Sie die Modellierung um eine “data warehousing”-Anwendung. Diese soll für jedes Buch im Katalog festhalten, mit welchen anderen Büchern es zusammen bestellt wurde, und wie oft. Wenn bei einer Bestellung eines Buches ein anderes, oft bestelltes Buch, noch nicht bestellt worden ist, soll das System dieses andere Buch vorschlagen.

12 *Ein generischer Webshop*

10 Punkte

Eigentlich ist die gesamte Logik relativ unabhängig von den verkauften Waren– nicht nur unser großes Vorbild in dieser Aufgabe hat erkannt, dass es eigentlich ziemlich egal ist, ob man nun Espressomaschinen oder Bücher verkauft. Skizzieren Sie eine Abstraktion zu einer generischen Anwendung, die beliebigen Waren verkauft.