

# 1. Projektaufgabe DBS-II

10. November 2003

Abgabe: 15.12.2003      Betreuer: Joos-Hendrik Böse  
[boese@mi.fu-berlin.de](mailto:boese@mi.fu-berlin.de)

## Diskursbeschreibung

Eine europaweit operierende Vermittlung von Ferienhäusern verfügt in verschiedenen Ländern über Filialen, welche Häuser an Kunden vermitteln. Der Mietvertrag wird direkt zwischen dem Besitzer des Hauses und dem Kunden geschlossen, die Ferienhausvermittlung erhält eine Provision von 10% pro Vermittlung.

Ein Hausbesitzer, der sein Haus vermieten möchte, lässt sein Haus in einer Filiale der Vermittlung registrieren und gibt an, wieviel er in den verschiedenen Saisons dafür haben möchte und in welchen Zeiträumen sein Haus z.B. wegen Eigenbedarf nicht vermietet werden kann.

Grundsätzlich können alle Häuser von allen Filialen vermittelt. Um den Kunden bei der Auswahl eines Hauses besser beraten zu können, werden die Häuser bei der Registrierung in verschiedene Kategorien eingeteilt. Beispiele für solche Kategorien sind *“Objekte in Stadtnähe”*, *“Kindgerechte Ferienobjekte”* oder *“Romantische Gemäuer”*. Ein Ferienhaus kann dabei in mehrere Kategorien fallen.

## 1 Vorhandene Systemlandschaft

Die Daten werden in einer zentralen relationalen Datenbank gehalten, auf welche die Mitarbeiter der Filialen direkt mit einem kleinen grafischen Visual Basic Client zugreifen. Das VB Frontend benutzt ADO, um direkt auf die Datenbank zugreifen zu können. Ein HTML Frontend welches in Perl DBI entwickelt worden ist, erlaubt es Kunden sich über das Internet im Katalog der Vermittlung zu informieren.

Die Daten sind im folgendem Schema organisiert, es werden hier nur die wichtigsten Attribute (Schlüsselattribute sind *fett* geschrieben) angegeben :

- Haeuser(**HausNr**, AnbieterNr, LandNr, Ort, Region, KurzB, ...)
- Anbieter(**AnbieterNr**, Anrede, Vorname, Name, Strasse, ...)
- Kategorie(**KategorieNr**, Beschreibung, ...)

- ErfuellteKategorie(*HausNr, KategorieNr*)
- Belegung(*HausNr, Beginn, Ende, Reserviert, ...*)
- Preise(*HausNr, SaisonNr, Preis, ...*)
- Saison(*SaisonNr, Name, Von, Bis, ...*)
- Filiale(*FilialNr, Ort, Land, Strasse, ...*)
- Buchungen(*HausNr, FilialNr, KundenNr, Von, Bis, Anzahlung, ...*)
- Kunde(*KundenNr, Anrede, Vorname, Name, Strasse, ...*)

Die kleine Firma kann es sich nicht erlauben, die Investition für das vorhandene RDBMS abzuschreiben. Dieses muß mit dem vorhandenen Schema weiterhin verwendet werden.

## 2 Aufgabenstellung

In Zukunft sollen alle Anwendungen sowohl für die Mitarbeiter der Filialen als auch für Kunden objektorientiert in Java entwickelt werden und dabei auf die gleichen Business Klassen zugreifen. Der IT Leiter erhofft sich dadurch, die Entwicklungen von neuen Anwendungen kostengünstiger und schneller realisieren zu können.

In einem ersten Entwicklungsschritt sollen die drei folgenden Anwendungsfälle mit objektorientierten Techniken implementiert werden:

1. Eine Filiale registriert ein neues Ferienhaus im System, welches einem schon registrierten Anbieter gehört.
2. Ein Interessent läßt sich im Onlinekatalog alle Häuser inkl. der Preise und Freizeiten anzeigen, die in einer bestimmten Region liegen und zu einer bestimmten Kategorie gehören.
3. Eine Filiale erzeugt eine Liste mit allen Buchungen für ein Haus.

Es soll eine objektorientierte Zugriffsschicht eingeführt werden, die es den Anwendungsentwicklern erlaubt Objektzugriffe auf Basis des Objektmodells vorzunehmen.

Zur Realisierung dieser Zugriffsschicht sollen die frei verfügbaren Objekt-Relationalen Mapping Werkzeuge *ObjectRelationalBridge* (OBJ)<sup>1</sup> sowie *Hibernate*<sup>2</sup> verwendet werden. Während OBJ die ODMG API sowie JDO unterstützt, verwendet Hibernate eine proprietäre API und Abfragesprache (*Hibernate Query Language* HQL).

Die Projektaufgabe soll in Gruppen bearbeitet werden, jede Gruppe soll dabei die Aufgabe mit Hilfe *eines* dieser Tools und *einer* Zugriffsart lösen. D.h. es gibt drei verschiedene Gruppen, die sich mit ODMG, JDO oder mit Hibernate beschäftigen.

<sup>1</sup>ObjectRelationalBridge <http://db.apache.org/objb>

<sup>2</sup>Hibernate <http://www.hibernate.org>

**Bearbeitungshinweise:** Das vorhandene Datenbankschema sowie Testdaten werden als SQL Export zur Verfügung gestellt. Weiterhin werden wir uns in der Übung auf ein gemeinsames Objekt Design der Business Klassen einigen, damit die Lösungen später vergleichbar sind. Focus des Projektes liegt auf dem zu implementierenden Objektrelationalen Mapping, sowie auf den unterschiedlichen Datenzugriffsmethoden (ODMG, JDO, Hibernate) zur Realisierung der drei Anwendungsfälle.