Übungsblatt

Ausgabe: 01.06.2001

In dem Praktikum soll ein System entwickelt werden, welches eine interaktive Verwaltung von Lehrveranstaltungen erlaubt.

Für jede Lehrveranstaltung sollen dabei folgende Daten verwaltet werden:

- Veranstaltungskennnummer (VAK),
- Titel,
- Veranstalter,
- Termine (Ort und Zeit), sowie
- der Inhalt der Veranstaltung als eine Liste von Themen.

  Zu jedem Thema einer Veranstaltung sollen Studenten Fragen stellen können, und Fragen anderer Studenten beantworten können. Diese Fragen und Antworten sollen für alle anderen Benutzer sichtbar sein.

Das System besitzt eine Client-Server-Architektur, und hat zwei Klassen von Nutzern: Studenten und Lehrveranstaltungsanbieter.

Der Client

Die Studenten interagieren nur mit den Client. Dieser sollte eine graphische Benutzerschnittstelle besitzen, und folgende Funktionen implementieren:

- Der Benutzer soll nach folgenden Kriterien nach Lehrveranstaltungen suchen können:
  - Titel
  - Veranstalter
  - Temin

  wobei eine Suche eine Konjunktion von Suchen nach Einzelkriterien ist, also zum Beispiel alle Veranstaltungen Dienstag morgen (8-12) von Bernd Krieg-Brückner.
- Das Ergebnis der Suche soll geeignet anzeigt werden, so dass der Benutzer eine der gefundenen Lehrveranstaltungen auswählen kann. Danach werden alle Daten der Lehrveranstaltung angezeigt; es bleibt der Implementation überlassen, ob alle Fragen und Antworten (und Notizen, s.u.) angezeigt werden, oder ob der Benutzer hier Fragen und Antworten zur Anzeige auswählen kann.

- An dieser Stelle kann der Benutzer neue Fragen stellen, oder zu bestehenden Fragen eine Antwort erstellen.
- Außerdem soll der Benutzer zu Themen private Notizen machen können. Diese Notizen sollen nur für den Benutzer sichtbar sein, und werden daher nur lokal auf dem Client (an definierter Stelle im Verzeichnis des Benutzers) gespeichert.

Der Server

Der Server verwaltet den zentralen Datenbestand, und implementiert folgende Funktionen:

- Suche nach bestimmten Lehrveranstaltungen. (Ein Sonderfall ist hierbei die Suche nach allen Lehrveranstaltungen.)
- Anzeige der Daten für eine Lehrveranstaltung.
- Änderung von Daten: eine neue Frage wird gestellt, oder eine neue Antwort gegeben.

Alle diese Funktionalitäten werden durch den Client genutzt. Der Server braucht also *nicht* die Erstellung neuer Lehrveranstaltungen zu unterstützen; diese wird extern durch das Lehrpersonal geleistet.

In welcher Form (als eine Datei, oder Sammlung mehrerer Dateine) der Server die Daten speichert, bleibt der Implementation überlassen. Allerdings ist das Format, in dem die Daten gespeichert werden, festgelegt (s.u.), und es muß auf jeden Fall sichergestellt werden.

Wichtig ist, daß der Server Anfragen nebenläufig behandelt, korrekt synchronisiert (zum Beispiel, wenn mehrere Studenten gleichzeitig Fragen stellen), und robust gegen Fehler auf Seiten der Clients ist. Der Server darf nicht ausfallen, wenn ein einzelner Client ausfällt.

Das Datenformat

Die Daten (Vorlesungen und Anfragen) werden in XML modelliert; die entsprechenden *document type definitions* (DTDs) finden sich im Anhang, und auf der Webseite.

In den Veranstaltungen der 29. Woche (12.06 und 14.06.) wir den Sax-Parser und HaXML vorstellen, die die einfache Verarbeitung von XML mit Java und Haskell erlauben.

Anhang: Document Type Definitions für Daten und Anfragen

1. Daten

Die folgende DTD beschreibt das Format für genau einen Kurs. Die VAK ist dabei als Attribut des Kurses kodiert, die anderen Daten als Elemente.

```
<!--
    Id:           kurs.dtd
    Autoren:      Hui Shi und Christoph Lueth
    Datum:        31.05.2001

    Beschreibung: Diese Datei gehoert zur Aufgabe des Kurses
                  "Internet Programmierung", und definiert der Typ
                  von Kursen.
-->
<!ELEMENT kurs (titel,veranstalter+,termin+,kommentar,thema+,diskussion)>
<!ATTLIST kurs key CDATA #REQUIRED>

<!ELEMENT veranstalter (name,vorname)>

<!ELEMENT termin (zeit,raum)>

<!ELEMENT titel (#PCDATA)>
<!ELEMENT zeit (#PCDATA)>
<!ELEMENT raum (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT vorname (#PCDATA)>
<!ELEMENT kommentar (#PCDATA)>
<!ELEMENT thema (#PCDATA)>
<!ELEMENT diskussion (frage*,antwort*)>
<!ELEMENT frage (#PCDATA)>
<!ATTLIST frage id CDATA #REQUIRED
                user CDATA #REQUIRED
                datum CDATA #IMPLIED>
<!ELEMENT antwort (#PCDATA)>
<!ATTLIST antwort ref CDATA #REQUIRED
                  user CDATA #REQUIRED
                  datum CDATA #IMPLIED>
```

Der Kurs Internet-Programmierung wird damit in etwa wie folgt dargestellt:

```
<kurs key="03-763">
  <titel>Internet----Programmierung</titel>
  <veranstalter><name>Shi</name><vorname>Hui</vorname></veranstalter>
  <veranstalter><name>Lueth</name><vorname>Christoph</vorname></veranstalter>
  <termin><zeit>Di 8-10</zeit><raum>MZH 7260</raum></termin>
```

```
  <termin><zeit>Do 9-10</zeit><raum>MZH 7260</raum></termin>
  <kommentar>Grundlagen der Internat-Programmierung</kommentar>
  <thema>Einführung: Grundkonzepte</thema>
  <thema>Nebenläufigkeit und Synchronisation</thema>
  <diskussion></diskussion>
</kurs>
```

Bei den Diskussionen besitzt jede Frage eine eindeutigen Bezeichner (Attribut `id`), und jede Antwort bezieht sich auf (der Einfachheit halber) eine Frage. Die eindeutigen Bezeichner für die Fragen müssen vom Server dynamisch generiert werden.

2. Anfragen

Eine Anfrage besteht aus einer EinzelAnfrage (Element `elem` unten), oder Verknüpfung von verschiedenen Einzelanfragen (Element `ausdruck`). Eine Einzelanfrage besteht aus einem Begriff, welches eines der möglichen Elemente eines Kurses entspricht (also `titel`, `veranstalter`, `key`, `thema`, `termin-zeit` und `termin-raum`), und dem Kriterium, welches den gesuchten Wert angibt. Die Verarbeitung von Suchmustern (regulären Ausdrücken usw.) ist optional.

Weiterhin werden die Elemente `neuefrg` und `neueant` definiert, die neu gestellte Fragen und Antworten entsprechen.

```
<!--
    Id:           anfrag.dtd
    Autoren:      Hui Shi und Christoph Lueth
    Datum:        31.05.2001

    Beschreibung: Diese Datei gehoert zur Aufgabe des Kurses
                  "Internet Programmierung", und definiert der Typ
                  von Anfragen.
-->

<!ELEMENT anfrag (elemt|ausdruck|neuefrg|neueantw)>

<!ELEMENT elemt (begriff,kriterium)>
<!ELEMENT begriff (#PCDATA)>
<!ELEMENT kriterium (#PCDATA)>

<!ELEMENT ausdruck (und)+>
<!ELEMENT und (elemt)*>

<!ELEMENT neuefrg (#PCDATA)>
<!ATTLIST neuefrg key CDATA #REQUIRED>

<!ELEMENT neueantw (#PCDATA)>
<!ATTLIST neueantw key CDATA #REQUIRED
                   ref CDATA #REQUIRED>
```