

LOTS - Online-Training an der Universität Leipzig

Timo Böhme, Erhard Rahm, Dieter Sosna

Universität Leipzig
Institut für Informatik

{boehme | rahm | sosna}@informatik.uni-leipzig.de
<http://dbs.uni-leipzig.de>

Zusammenfassung

Wir stellen ein Online-Übungssystem vor, das im Rahmen des Bildungsportals Sachsen entwickelt wurde und erfolgreich im Informatikstudium der Universität Leipzig im Einsatz ist. Der Systemzugang erfolgt für Lernende und Tutoren gleichermaßen über Web-Schnittstellen. Das System eignet sich insbesondere für den vorlesungsbegleitenden Übungsbetrieb, aber auch für das Selbststudium. Für die Ausbildung im Gebiet „Datenbanken“ wird sowohl ein SQL-Trainer als auch ein XQuery-Trainer zum Erlernen und Anwenden der jeweiligen Abfragesprache integriert. Wir stellen die Konzeption und Realisierung des Systems vor und berichten über bisherige Erfahrungen und geplante Erweiterungen.

1 Einleitung

Wir stellen das Online-Übungssystem LOTS (Leipzig Online Test System) [LOTS05] vor, das aus einem vom Freistaat Sachsen im Rahmen des Verbundprojektes „Bildungsportal Sachsen“ [BiPo05] geförderten Entwicklung entstanden und seit drei Jahren erfolgreich im Informatikstudium der Universität Leipzig im Einsatz ist. Wesentliche Zielsetzungen sind die Nutzung moderner Web-Technologien bei der Vertiefung des Lehrstoffes, insbesondere die Online-Präsentation und Online-Bearbeitung von Übungsaufgaben, Individualisierung der Aufgabenstellung und automatisierte Korrekturmöglichkeiten. Der Systemzugang erfolgt nicht nur für Lernende über Web-Schnittstellen mittels eines Web-Browsers, sondern auch für Tutoren und den Administrator.

Der Eigenentwicklung ging eine Analyse der zahlreichen vorhandenen Systemansätze [JOV03] voraus. Es zeigte sich, dass die meisten Ansätze eine administrativ und finanziell aufwendige Verwendung proprietärer Systementwicklungen verlangen. Weiterhin kann bei den zumeist kleineren Anbieterfirmen von keiner über mehrere Jahre gesicherten Systemlösung ausgegangen werden, was jedoch erforderlich wäre, um den Aufwand zur Erstellung der Lerninhalte in einem bestimmten Format zu rechtfertigen. Ein interessantes, von uns getestetes und mittlerweile kommerzialisiertes [Grad05]

Übungssystem zur webbasierten Bearbeitung von Multiple-Choice-Aufgaben wurde an der Universität Stanford entwickelt [Ullm03]. Die mit dem System gesammelten Erfahrungen sind in die Entwicklung von LOTS eingeflossen.

Die Implementierung und Nutzung unseres Systems erfolgte in zwei Phasen. In der ersten Phase wurden für die Ausbildung im Gebiet „Datenbanken“ das Üben der Anfragesprache SQL und ein elektronisches Tutorial für diese Sprache realisiert. Die Realisierung dieses Teilsystems mit der Bezeichnung „SQL-Trainer“ wird in [SR03] skizziert. In der zweiten Phase bis Ende 2004 wurde die LOTS-Kernfunktionalität zur Erstellung, Bearbeitung und Korrektur von Online-Übungsblättern realisiert. Seit Anfang 2006 ist auch eine Übungsmöglichkeit der XML-Anfragesprache XQuery integriert. Der Zugang zum Prototypen ist unter der URL <http://lots.uni-leipzig.de> möglich. Mit einem Gastzugang kann LOTS in gewissem Umfang von jedermann genutzt werden. Die Nutzung aller vorhandenen Möglichkeiten erfordert eine individuelle Anmeldung.

Im Folgenden präsentieren wir in Kapitel 2 eine Situationsanalyse des Übungsbetriebes, die Grundlage für die Konzeption des Systems war, welche wir, zusammen mit dem Systemaufbau, in Kapitel 3 beschreiben. In Kapitel 4 erläutern wir ausgewählte technische Details der Implementierung. Bisherige Erfahrungen beim Betrieb des Systems sind in Kapitel 5 dargestellt. In Kapitel 6 wird eine geplante Weiterentwicklung des Systems angesprochen. Kapitel 7 ist die Zusammenfassung.

2 Situationsanalyse und Anforderungen

Eine der traditionellen Lehrformen an Universitäten ist der vorlesungsbegleitende Übungsbetrieb. Durch das Stellen von Übungsaufgaben durch die Lehrkräfte erhalten die Studierenden eine Anleitung zum Selbststudium.

Eine Analyse des klassischen Übungsbetriebes ergab folgendes Ablaufschema:

- Ausgabe der Aufgabe und Bearbeitung durch die Studierenden, Abgabe der aufgeschriebenen Lösungen nach 1 bis 2 Wochen, meist noch auf Papier
- Korrektur/Auswertung durch wissenschaftliche Mitarbeiter oder durch studentische Hilfskräfte in durchschnittlich 3 bis 5 Tagen
- Erarbeitung einer Lösung bzw. Diskussion der studentischen Lösungen in den Übungsveranstaltungen

Bei diesem Ablauf fällt die stoßartige Arbeitsbelastung in der Korrekturphase auf, die proportional zu der Anzahl der Studierenden steigt. Die Erstellung von Lösungen auf Papier verursacht zudem logistische Nachteile bezüglich Abgabe, Aufbewahrung und Ausgabe der korrigierten Lösungen. Ungünstig ist ferner der lange Zeitraum zwischen der Erarbeitung der Lösung durch die Studierenden und der Besprechung. Wünschenswert sind dagegen eine kurzfristige Rückinformation über die Korrektheit einer Lösung und dass eventuell mehrere Ansätze zur Lösung verfolgt und verglichen werden können, so dass der Studierende dann die seiner Meinung nach beste Lösung einreichen kann.

Die Realisierung eines elektronischen Übungssystems kann diese Probleme in unterschiedlichem Umfang adressieren, wobei die Art der Übungsaufgaben

den Grad der Automatisierbarkeit wesentlich bestimmt. Die Bestandsaufnahme der bisher gestellten Übungsaufgaben zeigte, dass diese nach einigen Umformulierungen in verschiedene Klassen eingeordnet werden können.

Die wichtigsten dieser Klassen sind:

- Freitextaufgaben: Die Lösung ist vom Studierenden als frei zu formulierender Text anzugeben. Damit können beliebig komplexe Aufgaben gestellt werden, jedoch kann nur sehr eingeschränkt eine automatisierte Korrektur realisiert werden. Der wesentliche Vorteil liegt darin, dass die Lösungen elektronisch und nicht auf Papier erstellt werden und somit logistische Vorteile resultieren. Die Verwendung von Abbildungen oder Sonderzeichen in der Aufgabenstellung und in den Lösungen ist prinzipiell möglich, jedoch mit erhöhtem Aufwand verbunden.
- Lückentextaufgaben: Die Lösung besteht in der richtigen Auswahl (evtl. mehrfacher Auswahl) von Worten aus einem begrenzten Vokabular. Damit kann eine automatisierte Korrektur erreicht werden, jedoch ist der Einsatzbereich dieser Aufgabenform sehr eingeschränkt und allenfalls zum Testen einfachen Faktenwissens geeignet
- Multiple-Choice-Aufgaben
- Im Bereich der Informatikausbildung kommt dem praktischen Erlernen und Vertiefen von Programmiersprachen und interaktiven Datenbank-anfragesprachen eine wichtige Rolle zu. Charakteristisch für diesen Aufgabentyp ist eine hohe Komplexität der Lösung konkreter Aufgabenstellungen, wobei oft mehrere gültige Lösungen existieren. Eine automatische Korrektur solcher Lösungsvorschläge gestaltet sich schwierig. Eine Syntaxprüfung der vorgeschlagenen Anfragen kann (mit Einschränkungen) von einem Datenbankverwaltungssystem übernommen werden. Die semantische Prüfung kann unter bestimmten Voraussetzungen über den Vergleich der Antwortmengen mit einer Musterantwort erfolgen.

In der vorliegenden Version des Übungssystems werden bisher die beiden letztgenannten Aufgabentypen unterstützt. Insbesondere werden Multiple-Choice-Aufgaben in großer Flexibilität unterstützt, da sie zum einen eine vollautomatische Bewertung zulassen und zum anderen viele Aufgaben, die eigentlich einem anderen Aufgabentyp angehören, nach entsprechender Umformung auch als Multiple-Choice-Aufgaben formuliert werden können. So kann beispielsweise die Kenntnis einer Berechnungsvorschrift dadurch geprüft werden, dass der Übende eine Aufgabe selbst richtig ausrechnen muss, um die korrekte Antwort wählen zu können. An anderer Stelle können Aufgaben so gestaltet sein, dass zu Ihrer Lösung das Nachschlagen in der Literatur erforderlich wird. Solche langdauernden Lösungsprozesse sollten durch das Übungssystem gebührend unterstützt werden.

Eine weitere Anforderung ist die Individualisierbarkeit der Aufgabenstellung, d. h. von jeder Aufgabe können vom System mehrere Varianten generiert werden. Bei den Multiple-Choice-Aufgaben kann dies dadurch erreicht werden, dass vom Autor einer Aufgabe eine größere Zahl richtiger und falscher Lösungen vorgegeben werden. Aus diesen Vorschlägen wird für jeden Studenten eine

individuelle Aufgabe mit Hilfe einer Zufallsauswahl erzeugt und diesem angezeigt. Somit kann die eigenständige Arbeit der Studenten unterstützt werden.

Große Bedeutung kommt der einfachen Nutzbarkeit des Systems für alle Nutzergruppen zu, insbesondere durch die Studenten (Lernende) und Autoren. Dies erfordert einen einheitlichen Systemzugang über einen Web-Browser ohne die Notwendigkeit einer Installation von Zusatzsoftware.

3 Konzeption und Systemaufbau

Mit der derzeitigen Version von LOTS wird in erster Linie der vorlesungsbegleitende Übungsbetrieb unterstützt. Insbesondere sollten die elektronisch zu bearbeitenden Aufgaben Aufgabenblättern zugeordnet werden können, die typischerweise einem bestimmten Themenkomplex betreffen und in einer festgelegten Zeitspanne zu bearbeiten sind.

Alle im Übungsbetrieb agierenden Personen werden individuell unterschieden und einer Rolle zugeordnet. Unterstützte Rollen sind Tutor und Student sowie Administrator, der nicht direkt in den Übungsbetrieb eingreift, sondern technische Aufgaben bezüglich Hard- und Software wahrnimmt. Entsprechend der Rolle sind verschiedene Aktionen erlaubt. Einen Überblick der Aktionsmodule von LOTS zusammen mit den entsprechenden Rollen ist in Abbildung 1 gezeigt.

Die Arbeit der Studenten und der Tutoren erfolgt über ein intuitiv zu bedienendes Web-Interface, das alle benötigten Funktionen bereitstellt. Für den Tutor wurde insbesondere ein Aufgabeneditor integriert, der die Erstellung und die Änderung von Aufgaben ermöglicht. Mit seiner Hilfe kann ein Tutor die Aufgaben ohne Detailkenntnisse der technischen Datenstrukturen der Aufgaben erstellen, er kann sich auf die inhaltlichen Bestandteile der Aufgaben konzentrieren.

Im folgenden werden die Tätigkeiten, die den Rollen zugeordnet sind, genauer beschrieben.

3.1 Tutoren

Tutoren können Übungsaufgaben erstellen, Übungsgruppen einrichten, mehrere

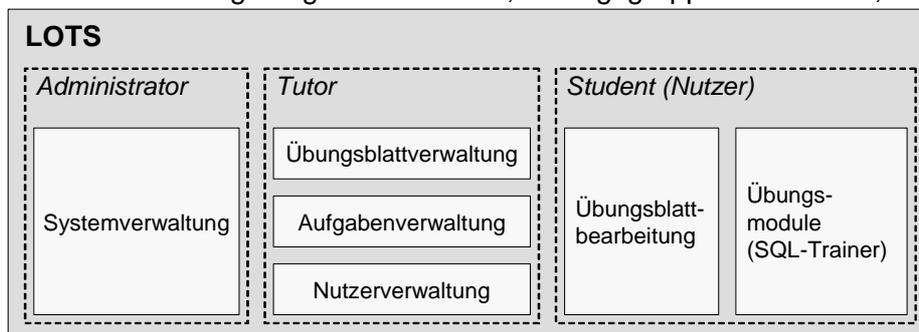


Abbildung 1: Systemkomponenten mit Rollenzuordnung

Aufgaben zu einem Übungsblatt zusammenstellen, Übungsblätter zur Bearbeitung für einen Zeitraum freigeben, die eingereichten Lösungen der Studenten und die Bewertung dieser Lösungen einsehen. Ferner können die Daten exportiert werden, um sie beispielsweise in einem Tabellenkalkulationsprogramm weiter zu verarbeiten.

Erstellen von Übungsaufgaben vom Multiple-Choice-Typ: Der Tutor gibt das Aufgabenthema, den Aufgabentext und möglichst viele verschiedene richtige und falsche Antworten vor (siehe Abbildung 2). Die Aufgabe kann so gestellt werden, dass erst nach einer intensiven Bearbeitung unter den angebotenen Lösungen eine sinnvolle Auswahl erfolgen kann.

Übungsgruppen einrichten: Der Tutor legt einen Namen für die Gruppe fest und ordnet diesem Namen ein Kennwort zu. Das Kennwort wird den Studenten beispielsweise in der Vorlesung genannt. Ein Student kann sich unter Nennung dieses Kennworts in die zugehörige Gruppe eintragen.

Übungsblätter zusammenstellen und Bewertungsschema festlegen: Nach dem Prinzip des Warenkorbes kann der Tutor in dem System vorliegende Aufgaben zusammentragen und diese Zusammenstellung als Übungsblatt ablegen. Weiter wird die Information abgefragt, wie viele Antworten dem Studenten pro Aufgabe vorgelegt werden sollen und wie viele davon richtig oder falsch sein können (Default: 4 Antworten mit mindestens einer richtigen und mindestens einer falschen Antwort). Diese Angaben werden auf Realisierbarkeit gegenüber der vorhandenen Menge von Antworten getestet.

The screenshot shows the 'Leipzig Online Test System' interface for creating a multiple-choice question. The page is titled 'Multiplechoiceaufgabe erstellen » Schritt 2 von 3'. The main content area is divided into several sections:

- Allgemeine Eigenschaften:**
 - Titel: LRU
 - Autoren: Erhard Rahm (kommagetrennte Liste)
- Bewertungsmaßstab:**
 - erreichbare Punkte: 2
 - Strafpunkte: 1
 - untere Punktegrenze: 0
- Frage (HTML zugelassen):**

Für einen mit LRU verwalteten Puffer der Seiten A, B, C und D referenziert und get Referenz auf D). Geben Sie an, ob nach genannten Referenzfolge die angegebene darstellt, d.h. sich an der untersten 1 befindet.
- Bemerkung (HTML zugelassen):**

[Empty text area]
- Antwortmöglichkeiten:**
 - Antwort 1: B, A, E, LRU-Element D (richtig?)
 - Antwort 2: B, A, B, LRU-Element C (richtig?)
 - Antwort 3: B, A, E, LRU-Element C (richtig?)
 - Antwort 4: B, A, E, LRU-Element E (richtig?)

Navigation buttons at the bottom include '< Zurück', 'Weiter >', 'Ab', and a secondary set of buttons: '< Zurück', 'Fertig', 'Übernehmen', 'Abbrechen'. A dropdown menu for 'zusätzliche Antwortfelder anzeigen:' is set to '1'.

Abbildung 2: Aufgabenerstellung

Leipzig Online Test System

UNIVERSITÄT LEIPZIG
Institut für Informatik
 Abteilung DBS

Logout
News
Verwaltung
Training
SQL-Training
XQuery
Profil
Impressum

Navigation

[zurück zur Übersicht](#)

System Info

boehme
#Benutzer: 3

13.10.2005
14:05:14

Universität Leipzig
 Institut für Informatik
 Abt. Datenbanken
 Prof. Dr. E. Rahm, T. Böhme

Ausgabe: 07.03.2005
 Abgabe: 09.03.2005
 Rückgabe: 20.03.2005

IDBS1 Kap. 3-4

Abgegebenes Übungsblatt, Versuch: 2 von 5, Wertung: 25% - 49%

Aufgabe 1 (Stacktiefenverteilung)

Für einen Puffer der Größe 4 seien zunächst die Seiten A, B, C und D gelesen worden. Bestimmen sie für die darauf folgenden Referenzen D, A, D, C, C, B zutreffende Kombinationen zwischen Stacktiefe und Trefferanzahl.

- Stacktiefe 1: 1 Treffer
- Stacktiefe 3: 2 Treffer
- Stacktiefe 4: 2 Treffer
- Stacktiefe 1: 2 Treffer

Aufgabe 2 (Adaptives LRU)

Für ein adaptives LRU-Verfahren seien bei einer Puffergröße c=4 gepuffert. Ferner sei p=1 und L1=B,F,E und L2=D,A,C,G, H. Geben Sie für die jeweils genannte Seitenreferenz an, ob die angegebene Auswirkung auf p

- Referenz auf D: Reduzierung von p
- Referenz auf G: keine Änderung von p
- Referenz auf A: keine Änderung von p
- Referenz auf C: keine Änderung von p

Ausgabe: 07.03.2005
 Abgabe: 09.03.2005
 Rückgabe: 20.03.2005

IDBS1 Kap. 3-4

Ihre Lösung, Versuch 2 von 5

Aufgabe 1 (Stacktiefenverteilung)

Für einen Puffer der Größe 4 seien zunächst die Seiten A, B, C und D gelesen worden. Bestimmen sie für die darauf folgenden Referenzen D, A, D, C, C, B zutreffende Kombinationen zwischen Stacktiefe und Trefferanzahl. 1 von 2

- Stacktiefe 1: 1 Treffer ✓
- Stacktiefe 3: 2 Treffer ✓
- Stacktiefe 4: 2 Treffer ✓
- Stacktiefe 1: 2 Treffer ✗

Aufgabe 2 (Adaptives LRU)

Für ein adaptives LRU-Verfahren seien bei einer Puffergröße c=4 die Seiten A,B,C und D gepuffert. Ferner sei p=1 und L1=B,F,E und L2=D,A,C,G, H. Geben Sie für die jeweils genannte Seitenreferenz an, ob die angegebene Auswirkung auf p zutrifft. 0 von 2

- Referenz auf D: Reduzierung von p ✗
- Referenz auf G: keine Änderung von p ✓
- Referenz auf A: keine Änderung von p ✗
- Referenz auf C: keine Änderung von p ✓

Bewertungsmaßstab	1 von 4
maximale Punkte pro Aufgabe:	2
minimale Punkte pro Aufgabe:	0
Strafpunkte:	1

Abbildung 3: Übungsblatt mit Bewertung in grober Stufe (links) und mit Lösung (rechts)

Ferner wird ein Bewertungsschema festgelegt. Dies enthält die Anzahl von Punkten bei richtiger Bewertung aller Antworten einer Aufgabe, die Anzahl von Strafpunkten pro falscher Teilantwort sowie die minimale Anzahl von Punkten pro Aufgabe. Im Beispiel von Abbildung 3 sind pro Aufgabe vier Antworten vorgegeben, von denen jede mit richtig oder falsch zu bewerten ist und mehr als eine richtige Antwort zugelassen ist. Werden alle 4 Antworten richtig bewertet, werden 2 Punkte erzielt, jede fehlerhafte Teilantwort führt zum Abzug von einem Punkt bis zum Minimum von 0 Punkten. Durch die Vergabe von Strafpunkten soll es sehr unwahrscheinlich werden, durch zufälliges Anklicken zu Punkten zu kommen.

Übungsblätter freigeben: Ein Übungsblatt wird einer oder mehreren Gruppen zugeordnet und mit Terminen zu Freigabe und Abgabe versehen. Außerdem kann eine maximale Anzahl von Übungsversuchen pro Blatt festgelegt werden. Innerhalb des Zeitraums zwischen beiden Terminen wird die Aufgabe den Mitgliedern der Gruppen nach einer Anmeldung im System zur Bearbeitung angeboten. Im Normalfall wird den Studenten, die eine Lösung eingereicht haben, die korrekte Lösung nach dem Abgabetermin angezeigt. Als Sonderform

kann aber auch die sofortige Anzeige der korrekten Lösung eingestellt werden, z.B. zur Klausurvorbereitung.

Lösung einsehen: Der Tutor kann ständig die eingereichten Lösungen und ihre Bewertung einsehen.

Daten exportieren und importieren: Der Tutor kann tabellarische Übersichten über seine Übungsgruppen als Textdateien ausgeben. Die Übungsaufgaben können als XML-Dateien exportiert bzw. aus solchen Dateien importiert werden. Daneben kann der Tutor einer Übungsgruppe aktuelle Mitteilungen im System einstellen, die den Mitgliedern der Gruppe beim Anmelden angezeigt werden.

3.2 Studenten

Studenten können sich für die Arbeit mit dem System registrieren, sich in einer oder in mehreren Gruppen anmelden und freigegebene Übungsblätter mit individualisierten Aufgaben bearbeiten. Für diese Bearbeitung haben sie eine festgelegte Anzahl von Versuchen (z.B. 5), deren Ergebnisse in einer Datenbank gespeichert werden. Nach Ablauf der Bearbeitungszeit kann ein Student seine letzte Lösung jedes Übungsblattes mit Bewertung einsehen (siehe Abbildung 3).

Registrierung und Anmeldung: Beim Registrieren füllt ein Student ein Formular mit seinen persönlichen Angaben aus. Er erhält daraufhin ein vorläufiges Passwort zugesandt. Mit dem von ihm gewählten Loginnamen und diesem Passwort erhält er Zugang zum System.

Das Anmelden in einer Gruppe geschieht dadurch, dass er die Gruppe aus einer Liste auswählt und das zugehörige Kennwort angibt. Danach ist er Mitglied dieser Gruppe und hat Zugriff auf die dort gestellten Übungsblätter sowie aktuelle Mitteilungen der Gruppe.

Bearbeiten von Übungsblättern: Wenn sich ein Student zu einer Sitzung anmeldet, werden ihm die für ihn aktuellen Übungsblätter zur Bearbeitung angeboten. Wählt er ein Übungsblatt aus, bekommt er den letzten Stand seiner Arbeit angezeigt. Hatte er dieses Blatt noch nicht bearbeitet, wird eine individuelle Variante des Übungsblattes mittels Zufallsauswahl generiert und in der Datenbank gespeichert. Andernfalls erhält er aus der Datenbank den letzten Zwischenstand seiner Arbeit.

Die Online-Bearbeitung eines Übungsblattes kann ohne Datenverlust unterbrochen werden, um beispielsweise die Lösungen erst zu erarbeiten. Nach der Bearbeitung der angebotenen Antworten und Abgabe eines gelösten Übungsblatts ermittelt das System sofort die Korrektheit der einzelnen Aufgaben. Standardmäßig erhält der Student das Ergebnis nur in einer groben Stufung mitgeteilt (z.B. zwischen 25-49% der erreichbaren Punkte), ohne jedoch die korrekte Lösung offen zu legen (siehe Abbildung 3). Damit weiß der Student, inwieweit noch weitere Versuche zur Verbesserung des Ergebnisses angezeigt sind. Die detaillierte Bewertung wird nach Ablauf der Bearbeitungszeit auf der Basis der letzten eingereichten Lösung dem Studenten angezeigt, wie sie auch der Tutor sieht (siehe Abbildung 3). Bei der erwähnten Sonderform (z.B. zur Klausurvorbereitung nach Ablauf der Vorlesungen) wird die Detailbewertung für jedes Übungsblatt unmittelbar nach der Einreichung angezeigt.

Für die Aufgaben ergibt sich damit der in Abbildung 4 gezeigte Lebenszyklus.

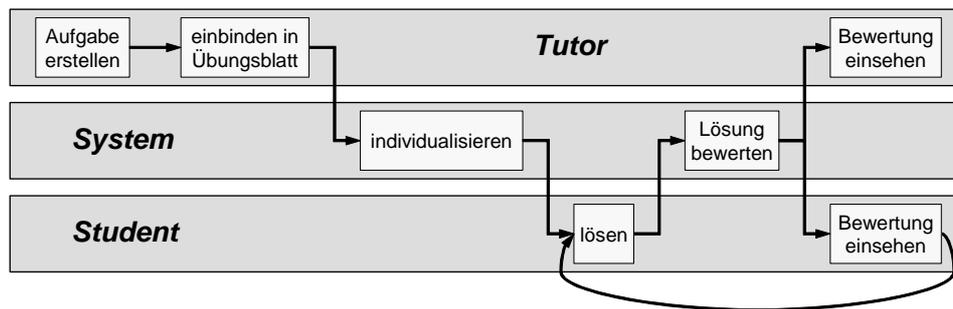


Abbildung 4: Lebenszyklus einer Aufgabe mit Rollenzuordnung

3.3 Administratoren

Der Administrator hat neben der Verwaltung der Tutoren eine mehr technische Funktion. Er ist verantwortlich für die Lauffähigkeit der Server-Hardware und der Software-Komponenten, insbesondere des Datenbanksystems, der Web- und Applikations-Server. Ferner ist er für die Datensicherung und eventuell auftretende Probleme bezüglich Zugriffsrechten, Email, etc. zuständig.

4 Zur technischen Realisierung

Eines der Design-Ziele für das Übungssystem war die weitgehende Unabhängigkeit von Hardware und Betriebssystem. Dies wurde durch Wahl der Programmiersprache JAVA erreicht. Für diese Sprache stehen viele qualitativ hochwertige Bibliotheken unter kostenfreien Lizenzen oder als Open-Source-Projekte zur Verfügung. Die Benutzung dieser Bibliotheken verkürzt die Entwicklungszeit und verringert die Fehlerhäufigkeit. Dennoch war noch ein großer Teil an Eigenentwicklung zu leisten. Durch einen modularisierten Aufbau und funktionale Abgrenzung der Bestandteile wurde die Grundlage für Erweiterungsmöglichkeiten geschaffen.

Die permanente Speicherung aller Daten, das betrifft sowohl die Nutzerdaten von Tutoren und Studenten wie auch Aufgaben, Lösungen und unterbrochene Bearbeitungen, erfolgt in relationalen Datenbanken mit Hilfe des freien Datenbankverwaltungssystems PostgreSQL. Dieses ist grundsätzlich durch ein anderes ersetzbar. Das Datenbankverwaltungssystem sichert die Persistenz.

Als Nutzerinterface dient, wie schon erwähnt, ein Web-Browser. Ein moderner Web-Browser erlaubt die Darstellung ohne weitere Zusatzsoftware (Plugins). Dieses erscheint uns wichtig, um die Akzeptanzschwelle für das System möglichst niedrig zu legen.

Ohne zu stark auf programmiertechnische Details einzugehen, soll beschrieben werden, wie die Erweiterung des Systems um neue Aufgabentypen erfolgen kann. Die Programmierung erfolgte streng objektorientiert. Für die Aufgaben wurde eine generische Klasse *Aufgaben* deklariert. Diese Klasse enthält als Attribut den Typ der Aufgabe und abstrakte Methoden für jede Aktion,

die für eine Aufgabe möglich ist, also z. B. für das Anzeigen einer Aufgabe, für das Speichern einer Lösung, für das Bewerten einer Lösung usw.

Für jeden Aufgabentyp, der im System vorkommt, ist in einem Aufgabenrepository die für diesen Typ spezialisierte Implementierung der genannten Methoden abgelegt. Soll nun zur Laufzeit mit einer Aufgabe eine beliebige Aktion ausgeführt werden, wird zunächst der Typ dieser Aufgabe an Hand des Attributs ermittelt. Dann werden die abstrakten Methoden durch die konkreten typspezifischen Methoden aus dem Repository ersetzt (überschrieben) und diese zur Ausführung gebracht.

Mit Hilfe dieser Konstruktion ist es möglich, das Übungssystem um neue Aufgabentypen zu erweitern:

- Die Methoden der Klasse *Aufgabe* sind für diesen Typ zu implementieren
- Die Implementierungen sind in das Repository einzutragen
- Der Name des neuen Aufgabentyps ist im System zu deklarieren.

Es sind somit keine Veränderungen an schon bestehenden Systemteilen erforderlich.

5 Erfahrungen

Das Üben der Anfragesprache SQL in LOTS (bzw. in dem ehemaligen SQL-Trainer) wird seit mehreren Jahren erfolgreich praktiziert, wobei auch zunehmend Nutzer außerhalb der Universität Leipzig zu verzeichnen sind (u.a. Kurseinsatz an Gymnasien). Seit 2003 wurde der elektronische Übungsbetrieb in mehrere Vorlesungen mit teilweise mehr als 200 Teilnehmern integriert. Wesentlicher Bestandteil des Integrationsprozesses war neben der Erstellung neuer Aufgaben die Transformation bestehender Übungseinheiten in die neue Form der Multiple-Choice-Aufgaben. Wo früher dem Studierenden nur eine Frage gestellt wurde, erhält er jetzt auch mögliche Antworten vorgelegt. Bei deren Formulierung war darauf zu achten, dass mit den Auswahlantworten möglichst keine Informationen über die Korrektheit der Antworten unterschwellig vermittelt wird.

Nach fast drei Jahren Benutzung kann festgestellt werden, dass das Angebot gut angenommen wird. Bezüglich der Klausurteilnehmer der jeweiligen Vorlesungen können folgende Aussagen getroffen werden:

- 60-70 % der Studierenden haben die Online-Übungen auf freiwilliger Basis vorlesungsbegleitend genutzt, davon haben ca. 90 % die Hälfte oder mehr der Übungsblätter bearbeitet
- bei 55 % der abgegebenen Übungsblätter nutzen die Studierenden die Möglichkeit der mehrfachen Bearbeitung
- Durchschnittlich bearbeitet ein Student ein Übungsblatt 2 - 3 Mal, auch die eingestellte Maximalzahl von 5 wird erreicht
- Studenten, welche LOTS zur Klausurvorbereitung nutzten, erzielten überdurchschnittlich gute Prüfungsergebnisse.

Die folgende Grafik soll diese Aussage unterstreichen, sie erhebt nicht den Anspruch einer wissenschaftlichen Studie, dazu wären systematische Untersuchungen nötig.

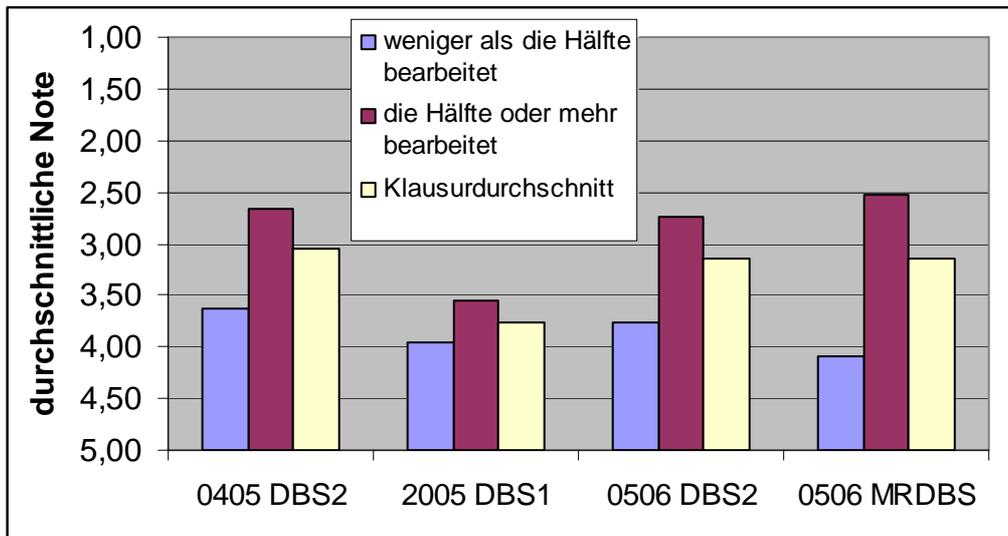


Abbildung 5: Durchschnittlich erreichte Note in verschiedenen Klausuren gruppiert nach der Bearbeitung der Online-Übungsaufgaben

Kritisch anzumerken ist, dass trotz der Individualisierung der Multiple-Choice-Aufgaben mittels zufällig ausgewählter Antworten ein hohes Maß an ‚Gruppenarbeit‘ zu beobachten war, wenn aus der Abgabe von Lösungen direkte Vorteile resultierten, wie z.B. anzurechnende Punkte bzw. Zulassungsvoraussetzungen. Diese moderne Form des Abschreibens wird speziell durch News-Foren und Chat-Systeme unterstützt. In einer unkontrollierten Umgebung wie sie hier vorliegt, in der die Studierenden zu jeder Zeit und von jedem Computer aus die Aufgaben bearbeiten können, lässt sich diese Zusammenarbeit nicht effektiv verhindern. Aus diesem Grund ist LOTS nicht zum prüfungsrelevanten Nachweis des Wissens vorgesehen bzw. geeignet, sondern primär zur besseren Vorbereitung auf die konventionell durchgeführten mündlichen Prüfungen bzw. Klausuren.

Am Beispiel dieser Implementierung soll noch ein wichtiges administratives Problem angesprochen werden. Wie jedes andere in der Lehre eingesetzte Mittel bedürfen auch die digitalen Hilfsmittel einer Pflege, ggf. einer Erneuerung, so auch Software. Hier sind nicht nur Fehler zu beseitigen oder inhaltliche Weiterentwicklungen vorzunehmen, sondern auch Weiterentwicklungen der Hardware und der Softwaretechnologie einzupflegen. Das ist ein Prozess, der über die gesamte Lebensdauer der Software andauert und der entsprechende personelle und finanzielle Ressourcen erfordert. Diese Tatsache wird bei Planungen häufig vernachlässigt. Es gibt u.E. auch keine allgemein gültige Lösungsformel für dieses Problem. Im vorliegenden Fall wurde durch das Bildungsportal Sachsen die angestrebte Dauerhaftigkeit der Resultate initiiert. Für internationale Projekte kann die Gestaltung als Open-Source-Projekt eine Nachhaltigkeit durch die gemeinschaftliche Pflege und Weiterentwicklung unterstützen. Summarisch kann eingeschätzt werden, dass der Einsatz computer- und internetgestützter Lehr- und Lernmittel nicht zu Einsparungen

führt, wohl aber zu qualitativen Verschiebungen. Es erfolgt in der Einsatzzeit solcher Systeme eine Entlastung der Mitarbeiter von Routinearbeiten, hin zu mehr kreativen, inhaltsbezogenen Tätigkeiten.

6 Weiterentwicklung zu einem offenen Übungssystem

Es gibt verschiedene Richtungen der Weiterentwicklung. Eine ist die Einbindung neuer Übungsmöglichkeiten, wie dies derzeit mit der Einbindung eines XQuery-Trainings mit Tutorial versucht wird. Eine andere ist die Erweiterung der Nutzergruppe. LOTS ist derzeit zwar nicht auf Studierende, jedoch auf Tutoren der Universität Leipzig beschränkt. Innerhalb eines neuen Projekts wurde daher mit der Entwicklung eines offeneren Übungssystems mit dem Arbeitsnamen O3 (Open Ontology-based Online training system) begonnen [Eck05]. Aufbauend auf LOTS entsteht hier ein System, welches in dreierlei Hinsicht offen ist:

- es ist offen für Online-Nutzer/Lernende, die sich an einem beliebigen Platz im Internet befinden
- es ist offen für alle Tutoren, die aus dem Internet heraus ihre Lehrinhalte einbringen möchten, wobei die Inhalte für andere freigegeben werden können
- es ist offen für Übungsaufgaben beliebiger Fachdisziplinen.

Die Erstellung guter Inhalte (hier Übungsaufgaben) noch immer eine zeitaufwendige Hauptaufgabe bei der Verwendung webbasierter Lernformen. Dieser Aufwand kann durch diverse gute Autorenwerkzeuge zwar gemildert werden, bleibt in jedem Falle aber beachtlich. Deshalb ist es ein Hauptanliegen von O3, die aufwendige Inhaltserstellung auf möglichst viele ‚Schultern‘ zu verteilen und die erstellten Inhalte einem breiten Nutzerkreis zugänglich zu machen. Neue Tutoren sollen durch die existierenden Inhalte sowie durch die Bereitstellung der Infrastruktur für das Übungssystem und für die Verwaltung der Studierenden für das System interessiert werden. Die Hoffnung ist, dass zumindest ein Teil der von den Tutoren neu eingebrachten Inhalte durch diese für andere Tutoren/Studenten freigegeben wird.

Die Lernenden sollen mit O3 ein einfach nutzbares Übungssystem für unterschiedliche Fächer und Fachrichtungen erhalten. Bedingt durch die Offenheit und den fachübergreifenden Einsatz wird auch ein neues Konzept für den Zugang zu den Inhalten benötigt. O3 bietet dem Studierenden deshalb drei Zugangsarten an

- *Navigation*
O3 enthält eine Ontologie (Taxonomie) der Fachdisziplinen und ihrer Untergruppen, denen die einzelnen Inhalte zugeordnet sind. Die Studierenden können darüber explorativ die verfügbaren Inhalte erschließen. Die Taxonomie wird nur in den beiden obersten Stufen vorgegeben und kann durch die Tutoren bedarfsgerecht erweitert werden.
- *Suche*
Durch die Vorgabe bestimmter Parameter, wie z.B. Fachgebiet, Tutor, Schwierigkeitsgrad oder Stichworte kann nach Inhalten gesucht werden und aus der Ergebnismenge Aufgaben zum Üben ausgewählt werden.

- *Übungsblätter*

Auch in O3 können Tutoren ausgewählte Aufgaben in Übungsblättern zusammenfassen, um diese einer Gruppe von Studierenden gezielt anzubieten. Die ausgewählten Aufgaben können auch von anderen Tutoren stammen (bzw. von Aufgaben anderer Tutoren abgeleitet worden sein), falls diese die Aufgaben zur allgemeinen Verwendung freigegeben haben.

Das Konzept von O3 basiert auf LOTS, verwendet jedoch ein erweitertes Rollenkonzept und zusätzliche Maßnahmen zur inhaltlichen Zuordnung der Aufgaben und Sicherheit des Systems. Eine erste Version von O3 wurde im Rahmen einer Diplomarbeit [Eck05] erstellt.

7 Zusammenfassung

Das vorgestellte Online-Übungssystem LOTS ist über gängige Web-Browser für Tutoren und Lernende nutzbar und unterstützt eine vorlesungsbegleitende Vertiefung und Selbstprüfung des Wissens und eine Vorbereitung auf Prüfungen. Durch geschickt ausgestaltete Multiple-Choice-Aufgaben können die Vorteile der automatisierten Lösungsbewertung auch für Aufgaben genutzt werden, deren korrekte Lösung die vorherige sorgfältige Bearbeitung durch die Studierenden verlangt. Das System wird bereits für mehrere Informatikvorlesungen an der Universität Leipzig genutzt und von den Studenten gut angenommen. Intensive Nutzer der Online-Übungsmöglichkeit schnitten in den Klausuren der betreffenden Lehrveranstaltungen überdurchschnittlich gut ab. Eine sinnvolle Weiterentwicklung von LOTS ist die Öffnung für unterschiedliche Fachdisziplinen und Tutoren.

Literaturverzeichnis

- [Eck05] Eckardt, J.: Konzeption und Implementierung eines offenen, ontologiebasierten Übungssystems. Diplomarbeit, Universität Leipzig, Juni 2005
- [JOV03] Jaeschke, P., Oberweis, A., Vossen, G.: Web basiertes Lernen: Eine Übersicht über Stand und Entwicklungen. In Web & Datenbanken (Hrsg.: E. Rahm, G. Vossen), dpunkt-Verlag 2003
- [SR03] Sosna, D.; Rahm, E.: Web-basiertes SQL-Training im Bildungsportal Sachsen. BTW-Workshop „Datenbanken und E-Learning“, Leipzig, Feb. 2003.
<http://dbs.uni-leipzig.de/en/projekte/BIPS/sqltr.pdf>
- [UI03] Ullman, J. D.: Improving the Efficiency of Database-System Teaching. Proc. ACM SIGMOD Int. Conference on Management of Data (invited talk), San Diego, 2003

Internetressourcen

- [LOTS05] Leipzig Online Test System.
<http://lots.uni-leipzig.de>
- [BiPo05] Bildungsportal Sachsen
Aus- und Weiterbildung der Hochschulen des Freistaates Sachsen
<http://www.bildungsportal-sachsen.de>
- [Grad05] Gradiance
<http://www.gradiance.com>