

1. Allgemeines

Das zu entwickelnde Produkt soll für den Übungsbetrieb an Hochschulen konzipiert werden. Die zu Grunde liegende Idee ist es, diesen weitestgehend zu automatisieren und für die Nutzer dieses Systems eine komfortable Schnittstelle auf Basis einer HTML-Seite zu bieten. Auf dieser HTML-Seite befindet sich ein Java-Servlet, mit dessen Hilfe der User seine Aktivität während des Übungsbetriebes abwickeln kann. Bei den drei existierenden Usergruppen handelt es sich um Studenten, MA und SHK, wobei MA der Mitarbeiter und SHK die studentische Hilfskraft symbolisiert. Zu Beginn erstellt der MA zu einer Lehrveranstaltung Übungsgruppen (Termin, Korrektor, Plätze) und stellt im Laufe des Semesters die Übungsaufgaben bereit. Nun erhält der Student die Möglichkeit sich erstens in eine Übungsgruppe einzutragen, falls noch ein freier Platz vorhanden ist, und zweitens die Möglichkeit, sich die aktuellen Aufgaben via Java-Applet herunterzuladen und seine Lösungen später einzusenden. Die eingesendeten Lösungen werden nun vom SHK heruntergeladen und korrigiert. Sobald dieses geschehen ist, trägt der SHK die erhaltenen Punkte jedes Studenten in eine MySQL-Datenbank ein. Diese Datenbank könne sowohl Studenten zur Einsicht ihres Punktestandes als auch der MA zum Erstellen von Punktübersichtslisten benutzen.

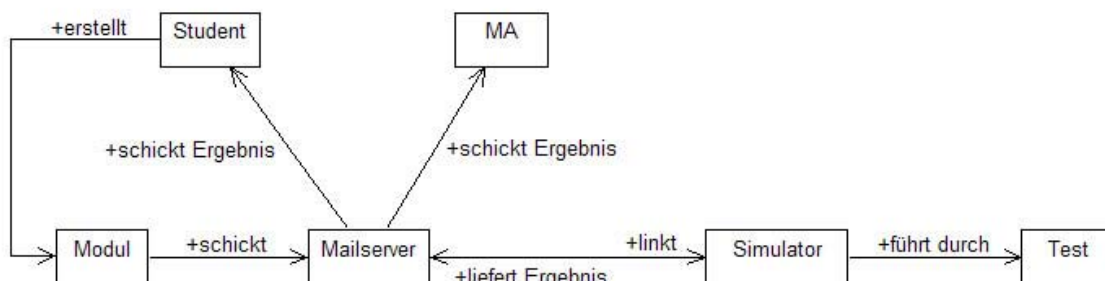
die wichtigsten Leistungsmerkmale:

allgemein:	Onlinefähigkeit Automatisierung Nutzung extern abgelegter Daten (Datenbank, XML) komfortable und einfache Bedienung
spezifisch für MA:	Erstellen von Übungsgruppen mit den dazugehörigen Daten Bereitstellung der Übungsaufgaben Punkteübersichtstabellenerstellung
spezifisch für SHK:	Herunterladen der zu korrigierenden Lösungsversuche Eingabe der erreichten Punktzahl in die Datenbank bzw. in eine XML-Datei
spezifisch für Student:	Anmeldung zu Übungsgruppen Abrufen der gestellten Aufgaben Upload der Lösungsversuche

2. Übersicht über themenrelevante Applikationen

2.1 Automata toolbox (autotool)

Hierbei handelt es sich um eine Applikation zur automatischen Korrektur von Übungsaufgaben der Abteilung Automaten und Sprachen an der Universität Leipzig. Autor der Applikation ist Johannes Waldmann, Informationen sind unter <http://theopc.informatik.uni-leipzig.de/~autotool/> zu finden. Der Umfang der zu korrigierenden Aufgaben ist mit (nichtdeterministischen) sequentiellen Automaten sehr begrenzt. Autotool ist in Haskell geschrieben, Übungsaufgaben selbst werden ebenfalls in Haskell implementiert. Diesen Haskell-Bytecode, bestehend aus Modul (Automat) und Varianten (Eingabe), schickt der Student an einen Mailserver, ein verlinktes Simulatorprogramm testet die Aufgaben und liefert das Ergebnis zurück. Student und / oder MA werden durch den Mailserver über das Ergebnis informiert.



Module: gängige, nichtdeterministische, sequentielle Automaten wie z.B. Turingmaschine, endlicher Automat, Kellerautomat in verschiedenen Sprachen, mit Varianten

Simulator: Simulatorprogramm, ermittelt alle möglichen Rechnungen (Lösungen) durch Breitensuche im Berechnungsbaum

2.2 WebAssign

Bei der Applikation WebAssign der FernUniversität Hagen, Lernraum Virtuelle Universität, handelt es sich um ein eigenständiges Anwendungssystem mit Zugriff über eine Webschnittstelle. Es setzt auf einer Servlet-Architektur auf, ist objektorientiert in Java implementiert und besitzt weiterhin Schnittstellen zu relationalen Datenbanken wie MySQL. Informationen sind unter <http://niobe.fernuni-hagen.de/WebAssign/> zu finden. MA planen Lehrveranstaltungen und richten entsprechend eine Webschnittstelle, also eine HTML-Seite mit Informationen, Übungsaufgaben, Musterlösungen ein. Studenten rufen Aufgaben ab und senden ihre Lösungen ein, welche von den SHK gemäß der Musterlösung korrigiert, bewertet und zurückgeschickt werden. Auf diese Applikation wird im folgenden noch näher eingegangen.

3. Genauere Beschreibung einer Applikation, hier: WebAssign

3.1 Verbale Beschreibung der Leistungsmerkmale

Es handelt sich um ein Anwendungssystem zur Unterstützung und Durchführung des Übungsbetriebes. Basierend auf einer Java-Servlet-Architektur als zentrale Schnittstelle wurde WebAssign objektorientiert in Java JDK 1.2 implementiert und unterstützt die Daten- bzw. Dokumentenverwaltung mit Hilfe relationeller Datenbanken wie MySQL oder Informix. Als Benutzerschnittstelle fungiert eine, durch den MA eingerichtete, statische HTML-Seite mit z.B. Anmeldung, Übungsaufgaben, Punktestand. WebAssign ist lauffähig unter Solaris 2.6 / 2.8 oder Linux in Verbindung mit einem Apache-Webserver und zusätzlichem MySQL- oder Informix-Datenbankserver. Weiterhin existieren Schnittstellen zu CORBA und JDBC. MA richten Lehrveranstaltungen ein, bieten also Anmeldungen, Übungsaufgaben und Lösungen an. Weiterhin werden im Folgenden Bearbeitungszeiten für einzelne Aufgaben gesetzt, der Korrekturmodus sowie die Quittierung festgelegt. Studenten können sich anmelden, Aufgaben abrufen und ihre Lösungen einsenden (ggf. mehrfach), die bei erfolgreicher Abgabe quittiert werden. SHK können sowohl Aufgaben, als auch Lösungsversuche und Musterlösungen der MA abrufen. Nach der Korrektur, die wahlweise online oder offline geschehen kann, wird diese noch einmal durch den MA geprüft, danach per E-Mail quittiert. Für den MA, als auch für die Studenten ist es möglich, jederzeit Informationen über Punkte, Bewertungen oder Korrekturen abzurufen.

3.2 Beschreibung von Anwendungsfällen

Anwendungsfall	Webschnittstelle erstellen
[evtl. Akteur]	MA
Beschreibung	der MA richtet eine statische HTML-Seite mit Anmeldung, Aufgaben usw. ein
Ergebnis	Studenten können sich zu Übungen anmelden und Informationen abrufen

Anwendungsfall	Übungsaufgaben / Musterlösungen erstellen
[evtl. Akteur]	MA
Beschreibung	der MA stellt auf der von ihm eingerichteten HTML-Seite Aufgaben (Studenten) und Lösungen (SHK) bereit
Ergebnis	Studenten können Aufgaben abrufen und bearbeiten

Anwendungsfall	Übungsaufgaben abrufen / bearbeiten
[evtl. Akteur]	Student
Beschreibung	der Student ruft verfügbare Aufgaben im Internet ab und bearbeitet diese
Ergebnis	Studenten können Aufgaben bearbeiten und Ergebnisse einsenden (ggf. mehrmals)

Anwendungsfall	(studentische) Lösungen einsenden
[evtl. Akteur]	Student
Beschreibung	der Student lädt seine Lösungsversuche hoch
Ergebnis	(studentische) Lösungen stehen dem SHK zur Korrektur bereit

Anwendungsfall	(studentische) Lösungen / Musterlösungen abrufen
[evtl. Akteur]	SHK
Beschreibung	der SHK ruft die Lösungen der Studenten und des MA zur Korrektur ab
Ergebnis	(studentische) Lösungen sind korrigiert und uploadfähig

Anwendungsfall	(studentische) Lösungen einsenden
[evtl. Akteur]	SHK
Beschreibung	der SHK sendet die korrigierten Lösungen der Studenten ein und trägt ggf. Punkte und Bewertungen ein
Ergebnis	Bewertung der Lösungen für MA verfügbar

Anwendungsfall	Überprüfung und Freigabe der (studentischen) Lösungen
[evtl. Akteur]	MA
Beschreibung	MA gibt korrigierte Lösungen / Bewertungen an Studenten frei
Ergebnis	Übungsaufgabenserie abgeschlossen

3.3 Diagramme

