



Übungen im Internet

Benutzerhandbuch

Stand: Dezember 2001

FernUniversität Hagen

Lehrgebiet Praktische Informatik III -- Software Engineering

Copyright © 2001 CampusSource, <http://www.campussource.de>

Inhalt

1	Einführung.....	4
1.1	Zu diesem Dokument	4
1.2	Über WebAssign	4
1.3	Softwaretechnische Voraussetzungen.....	5
1.4	Leitfäden für Studierende und Korrektoren	5
2	Zugang zum WebAssign-System	5
2.1	Allgemeines	5
2.2	Zugang zu Kursen der FernUniversität.....	6
2.3	Individuelles Anmelden von Studierenden	6
2.4	Anmeldung von Betreuern und Korrektoren	6
2.5	Wechsel der Identität	6
2.6	Betreuer und Korrektor in Personalunion	7
2.7	Spezielle Benutzerkennungen	7
3	Die WWW-Oberfläche des WebAssign-Systems	7
3.1	Wichtige Seiten des Systems	7
3.2	Die Kursstartseite	8
3.3	Die Studentenstartseite.....	8
3.4	Aufgabenseite.....	8
3.5	Quittung	9
3.6	Persönliche Ergebnisse	9
3.7	Persönliche Kursmappe.....	9
4	Funktionen für Betreuer Teil I: Betrieb	9
4.1	Die Betreuerstartseite	9
4.2	Aufgabenübersicht	10
4.3	Aufgabenkonfiguration	10
4.4	Der Aufgabenerstellungs-Assistent.....	11
4.5	Kursressourcen-Upload.....	13
4.6	Kursressourcen-Verzeichnis	14
4.7	Kursressourcen als Zip-Datei	14
4.8	Korrekturmodul-Verzeichnis	14
4.9	Einsendungs- und Korrekturübersicht	14
4.10	Einsendungen und Korrekturen eines Korrektors und eines Aufgabenheftes	15
4.11	Typischer Ablauf für die Arbeit eines Betreuers mit Einsendungen	15
4.12	Korrekturzipdateiverwaltung Alte Vorgehensweise - Bitte nicht mehr verwenden!!! ..	15
4.13	Erneute Hefteinsendung	16
4.14	Statistik.....	16
4.15	Belegerliste.....	16
4.16	Studenten anmelden	16
4.17	Studenten bearbeiten	16
4.18	Studentenliste	16
4.19	Betreuerliste	17
4.20	Betreuer anmelden	17
4.21	Betreuer bearbeiten	17
4.22	Korrektorenliste	17

4.23	Korrektor anmelden	17
4.24	Korrektor bearbeiten	17
4.25	E-Mail an Kurs-Beleger versenden	17
4.26	Message of the Day definieren.....	17
4.27	Kursparameter definieren.....	17
5	Funktionen für Betreuer Teil II: Aufgabenseiten.....	18
5.1	Teilaufgaben.....	18
5.2	Eingabefelder	19
5.3	Knopf zum Einsenden	19
5.4	Applets zur Erweiterung der Eingabemöglichkeiten	19
5.5	Einsenden von lokal erstellten Dateien	21
5.5.1	Webbasierter Ansatz vs. externe Dateiformate	21
5.5.2	Aufgabenseiten	22
5.5.3	Spezielle Variablen	22
5.6	Konsistenz von Aufgabenseiten, Quittungen und Korrekturschablonen	23
5.7	Heraufladen von HTML-Seiten	23
5.8	Zum Verständnis der WebAssign-URLs:	24
5.9	Bitmaps, Applets und zusätzliche HTML-Seiten	24
5.10	Heraufladen von Kursressourcen in Zip-Dateien.....	24
5.11	Links auf andere Seiten des WebAssign-Systems	25
5.12	Links auf Quittungsseiten	25
5.13	Probleme mit falschen Links und nicht dargestellten GIFs	25
5.14	Änderbare Seiten.....	25
5.15	Variablen.....	26
6	Funktionen für Betreuer Teil III: Korrekturen.....	28
6.1	Vorkorrektur.....	28
6.2	Korrektur	28
6.3	Abläufe und Überprüfung der Korrekturmechanismen	29
6.4	Der Lebenszyklus von Handkorrekturen	29
6.5	Automatische Korrektur über Korrekturserver	30
6.5.1	Kursspezifische KServer	31
6.5.2	Vordefinierte KServer	33
7	Funktionen für Korrektoren.....	34
7.1	Online-Korrektur.....	34
7.1.1	Korrektorstartseite	34
7.1.2	Einsendungen und Korrekturen eines Aufgabenhefts	34
7.1.3	Korrektur einzelner Aufgaben	35
7.2	Offline-Korrektur	35
7.2.1	Korrektorstartseite	35
7.2.2	Installation und Start des lokalen Servers	36
7.2.3	Einstellungen der Software für Korrektoren	36
7.2.4	Herunter- und Heraufladen von gepackten Korrekturen	36
7.2.5	Einsendungen und Korrekturen eines Aufgabenhefts	37
7.2.6	Zustände bei der offline Korrektur	37
7.2.7	Korrektur einzelner Aufgaben	37

1 Einführung

1.1 Zu diesem Dokument

Das vorliegende Handbuch richtet sich an die drei Benutzergruppen von WebAssign: Betreuer, Studierende und Korrektoren. Während Studierende und Korrektoren jeweils nur einen kleinen Ausschnitt der WebAssign-Funktionen nutzen, die zudem überwiegend intuitiv zu benutzen sind, erfordern die Funktionen für die Betreuer eine ausführlichere Einführung. Dieses Handbuch soll vor allem die Betreuer von Übungsveranstaltungen unterstützen. Der letzte Abschnitt dieser Einleitung enthält daher Leitfäden, die Studierenden und Korrektoren einen besseren Überblick über die jeweils relevanten Kapitel bzw. Abschnitte geben.

Im Wesentlichen behandelt dieses Handbuch die folgenden Themen:

- **Zugang zum WebAssign-System.**
Wie gelangen die unterschiedlichen Benutzer ins System?
Siehe Zugang zum WebAssign-System.
- **Die WWW-Oberfläche des WebAssign-Systems.**
Welche Seiten hat das WebAssign-System und was kann man dort machen?
Siehe Die WWW-Oberfläche des WebAssign-Systems.
- **Betrieb des WebAssign-Systems.**
Welche Aufgaben fallen für den Betreuer im laufenden Betrieb von WebAssign an?
Siehe Funktionen für Betreuer Teil I: Betrieb.
- **Erstellung und Pflege von Aufgabenseiten.**
Wie erstellt und pflegt man Aufgabenseiten für das Internet?
Siehe Funktionen für Betreuer Teil II: Aufgabenseiten.
- **Behandlung von Korrekturen.**
Was geschieht mit den Einsendungen der Studierenden?
Siehe Funktionen für Betreuer Teil III: Korrekturen.
- **Korrigieren mit Hilfe von WebAssign.**
Wie unterstützt WebAssign die Korrektoren?
Siehe Funktionen für Korrektoren.

1.2 Über WebAssign

WebAssign ist eine webbasierte Softwarekomponente, die den universitären Übungsbetrieb in das Konzept der virtuellen Universität integriert. WebAssign unterstützt das Erstellen von Aufgaben, das Herunterladen der Aufgabenstellungen auf den studentischen PC, das Einsenden der studentischen Lösungen, die Verteilung der Lösungen an die Korrekturkräfte, die Korrektur und Bewertung der Lösungen sowie das Zurücksenden der Korrekturen nebst der Musterlösungen an die Studierenden. In einer Datenbank werden sämtliche Aufgabenstellungen und Musterlösungen sowie die studentischen Lösungen, Korrekturen und Bewertungen gespeichert.

WebAssign erlaubt unterschiedliche Arten von Aufgabentypen und Korrekturmodi:

- Multiple-Choice Aufgaben, Zuordnungsaufgaben und numerische Aufgaben, die automatisch korrigiert und bewertet werden,
- offene Aufgaben, die manuell korrigiert und bewertet werden, und
- Programmieraufgaben, die innerhalb eines bestimmten Zeitraums wiederholt einem Online-Vortest unterzogen werden können, bevor sie abschließend manuell bewertet werden.

Die Entwicklung des Systems ist vom Universitätsverbund Multimedia im Rahmen der Ausschreibung *Neue Medien in der Hochschullehre* gefördert worden (<http://www.uvm-nw.de>). Weitere Informationen gibt es auch auf der Projekt-Homepage (<http://www-pi3.fernuni-ha>

gen.de/WebAssign).

1.3 Softwaretechnische Voraussetzungen

Im WebAssign-System greifen alle beteiligten Nutzer über eine Web-Schnittstelle auf ein zentral installiertes Anwendungssystem zu. Dort werden alle anfallenden Daten und Dokumente in einer Datenbank verwaltet. Als Benutzungsschnittstelle für WebAssign sind herkömmliche Web-Browser zu verwenden, die den folgenden Anforderungen genügen müssen:

Für Studierende: Von Seiten des WebAssign-Systems kommen nur sehr geringe Anforderungen an den verwendeten Browser. Jeder derzeit gebräuchliche Browser sollte dazu geeignet sein. Erprobt sind die Version 3 und 4 von Netscape und Internet Explorer. Allerdings werden durch die von den Veranstaltern erstellten Seiten, insbesondere die Aufgabenseiten, ggf. zusätzliche Anforderungen erzeugt, etwa Frames, JavaScript, Java, HTML 4 usw. Kursbetreuer sollten ihre Studierenden über solche Anforderungen informieren.

Für Betreuer und Korrektoren: Kritische Anforderung an den Browser von Betreuern und Korrektoren ist die Fähigkeit, neu erstellte oder geänderte HTML-Seiten über PUT-Requests an den WebAssign-Server zu übermitteln. Derzeit sind dazu nur die Editoren von Netscape in der Lage: Netscape Gold bzw. Composer. Dabei ist die Arbeit mit dem Composer wesentlich eleganter und einfacher. Neuere Versionen des Internet Explorers sind ebenfalls in der Lage mit dem Netscape Composer zusammenzuarbeiten.

1.4 Leitfäden für Studierende und Korrektoren

Leitfaden für Studierende:

- Zugang zum WebAssign-System - Abschnitte 2.1 bis einschließlich 2.3
- Die WWW-Oberfläche des WebAssign-Systems - Abschnitte 3.1 bis einschließlich 3.7
- Die Behandlung von Korrekturen - Abschnitte 6.1 und 6.2

Leitfaden für Korrektoren:

- Zugang zum WebAssign-System - Abschnitte 2.1 bis einschließlich 2.4
- Die WWW-Oberfläche des WebAssign-Systems - Abschnitte 3.1 bis einschließlich 3.7
- Die Behandlung von Korrekturen - Abschnitte 6.1 und 6.2
- Funktionen für Korrektoren - Kapitel 7

2 Zugang zum WebAssign-System

2.1 Allgemeines

Betreuer, Studierende und Korrektoren haben eine auf die entsprechende Benutzergruppe angepasste Sicht auf das WebAssign-System. Zur Gewährleistung personalisierter Zugänge zu dem System ist es erforderlich, dass sich jeder Benutzer durch Angabe einer Benutzerkennung und einem Passwort ausweist. Deshalb sind nur wenige Seiten öffentlich zugänglich, von denen aus der Zugriff auf die geschützten Bereich bereitgestellt wird. Nach der Authentifizierung darf ein Benutzer nur soweit Informationen sehen und verändern, wie dies für seine Aufgaben erforderlich ist. Insbesondere die persönlich konfigurierten Seiten, z.B. die persönlichen Ergebnisse eines Studierenden oder die zu korrigierenden Einsendungen eines Korrektors, sind nur für die entsprechende Person zugänglich.

Für jeden Veranstalter wird anfänglich eine Betreuer-Benutzerkennung mit Passwort eingerichtet. Mit dieser Benutzerkennung ist es möglich, weitere Betreuer und Korrektoren sowie ggf. Studierende anzumelden. In der Regel verwendet man den (Vor- oder Nach-) Namen als Benutzerkennung.

Für die Anmeldung von Studierenden gibt es verschiedene Möglichkeiten, die in den beiden folgenden Abschnitten erläutert werden.

2.2 Zugang zu Kursen der FernUniversität

Der Standardweg für Kurse der FernUniversität sieht vor, dass Studierende sich beim Universitäts-Rechenzentrum (URZ) einen allgemeinen Zugang zum FernUni-Netz geben lassen. Die Benutzerkennung lautet dabei immer $q<Matrikelnr>$. Mit dieser Kennung und dem zugehörigen Passwort kommen die Studierenden dann in alle Kurse, die sie aktuell belegt haben. Eine eigene Anmeldung gegenüber dem WebAssign-System ist hier nicht erforderlich.

2.3 Individuelles Anmelden von Studierenden

Für Kurse anderer Universitäten kann entweder ein Kursbetreuer „seine“ Studierenden von Hand anmelden oder er stellt eine öffentliche Anmeldungsseite zur Verfügung, über die sich die Studierenden selbst anmelden können. Das individuelle Anmelden einzelner Studierender ist aber auch für Kurse der FernUni noch möglich.

Zur Anmeldung eines Studierenden sind Name, Matrikelnummer und E-Mail-Adresse erforderlich. Die Matrikelnummer muss je Veranstalter eindeutig sein. Sie dient als Benutzerkennung bei der Prüfung der Zugangsberechtigung. Das zugehörige Passwort wird automatisch generiert und per E-Mail zugeschickt. Hierzu ist es erforderlich, dass die E-Mail-Adresse korrekt ist! Name und E-Mail-Adresse können nachträglich geändert werden (nicht die Matrikelnummer).

Die Anmeldungsseite gehört immer zu einem bestimmten Kurs. Ein neu angemeldeter Student wird gleichzeitig als Beleger dieses Kurses eingetragen und in die Studentenliste des Veranstalters aufgenommen.

Ist ein Studierender bereits in der Studentenliste des Veranstalters eingetragen (weil er schon einmal einen Kurs belegt hat), so braucht man zur Belegung nur seine Matrikelnummer anzugeben.

Die öffentliche Anmeldung funktioniert analog.

2.4 Anmeldung von Betreuern und Korrektoren

Die Anmeldung von Betreuern und Korrektoren erfolgt analog zum individuellen Anmelden von Studierenden.

2.5 Wechsel der Identität

Zum Einrichten und Testen einer Übungsveranstaltung ist es erforderlich, sowohl als Betreuer als auch als Korrektor oder Student auf das System zuzugreifen, um die einzelnen Seiten bzw. Funktionalitäten anzusehen bzw. ausführen zu können. Allerdings ist ein Wechsel der Identität nicht direkt möglich: Ein Browser merkt sich die zuletzt verwendete Autorisierung und benutzt diese so lange, bis der Server sie zurückweist. Dann erst kann man Benutzerkennung und Passwort neu eingeben. Alternativ kann man den Browser zweimal starten oder zwei verschiedene Browser starten. In einem agiert man als Betreuer, im anderen als Student.

Einige Links ermöglichen bzw. erzwingen einen Wechsel der Identität. Am wichtigsten sind hierbei die Links auf der Startseite eines Kurses. Der Link *Alle Betreuungsfunktionen* erzwingt die Autorisierung als Betreuer. Analog erzwingen die Links *Korrektorübersicht* und *Ihre persönliche Aufgabenübersicht* eine Identifikation als Korrektor bzw. Studierender. Ein weiterer nützlicher Identitätswechsel findet statt, wenn man als Betreuer auf eine Aufgabenseite gekommen ist. Drückt man auf einen Knopf zum Einsenden kommt zunächst eine Warnung, dass man kein Student ist. Anschließend kann man sich als Student identifizieren.

2.6 Betreuer und Korrektor in Personalunion

Wenn man eine Person als Betreuer und als Korrektor jeweils mit identischem Namen und identischem Passwort einträgt, kann man nahtlos, das heißt ohne Passwortabfrage, zwischen Betreuer- und Korrektorseiten wechseln.

2.7 Spezielle Benutzerkennungen

Gastzugang Mit der Benutzerkennung *gast* und dem Passwort *webassign* kann man die Aufgabenseiten aller Veranstalter ansehen (alle anderen Benutzerkennungen gelten jeweils nur für den Veranstalter, der sie eingerichtet hat).

Teststudent Ein Student mit der speziellen Matrikelnummer *7777777* kann unabhängig von Bearbeitungsfristen auf Aufgabenseiten und Musterlösungen zugreifen. Insbesondere kann man mit Hilfe dieser Benutzerkennung Aufgabenseiten testen, ob z.B. die Quittungsseiten korrekt sind, eine evtl. eingerichtete Vorkorrektur funktioniert usw. Den Teststudenten muss jeder Veranstalter selbst anmelden, insbesondere ein eigenes Passwort vergeben.

Demobetreuer Ein Betreuer mit dem Namen *demo* hat lesenden Zugriff auf Betreuerseiten, darf dort allerdings keine verändernden Funktionen aufrufen und keine Benutzerdaten (insbesondere Passwörter) ansehen. Wie beim Teststudenten muss ein Demobetreuer von jedem Veranstalter selbst angemeldet werden.

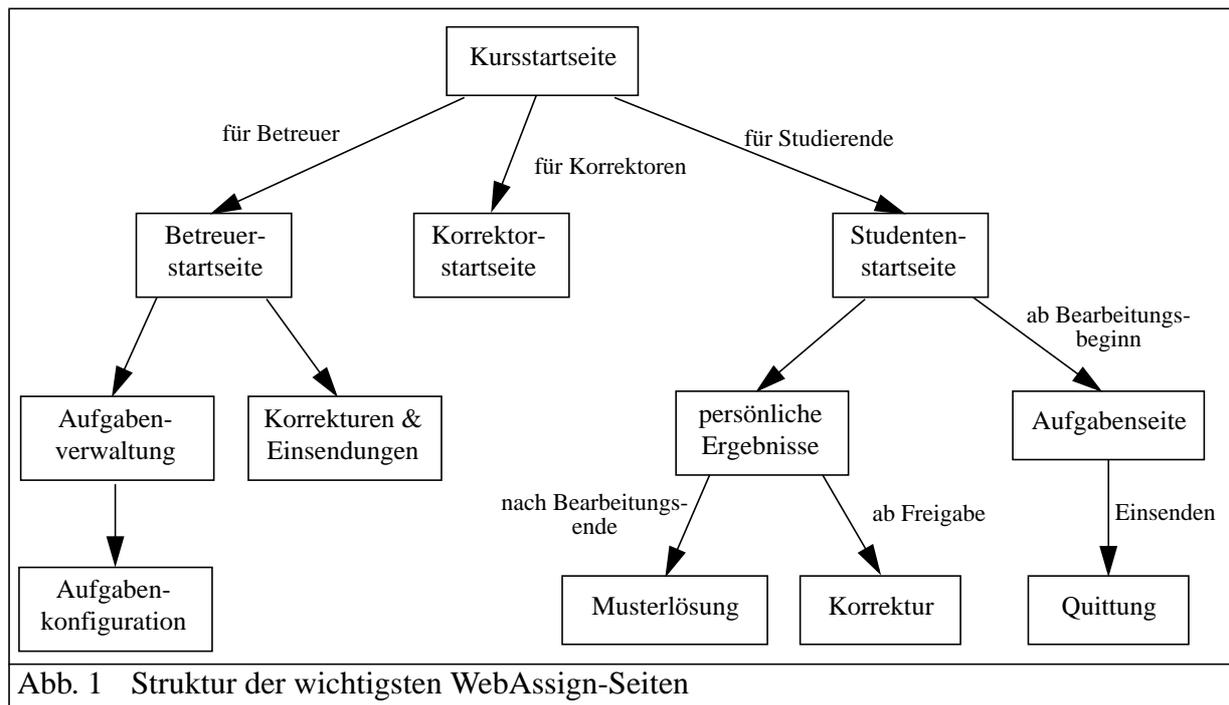
Betreuer als Student Manchmal ist es hilfreich, wenn man als Betreuer die Identität eines Studenten annehmen kann, um die Probleme, die ein Student beschreibt, direkt nachvollziehen zu können. Zu diesem Zweck dient eine Benutzerkennung, die aus der Betreuerkennung, einem angehängten '@' und der Matrikelnummer des Studierenden besteht. Als Passwort wird das zugehörige Betreuerpasswort verwendet.

3 Die WWW-Oberfläche des WebAssign-Systems

3.1 Wichtige Seiten des Systems

Die folgende Abbildung gibt einen Überblick der wichtigsten Seiten des Webassign-Systems. Jeder Kurs hat eine Einstiegsseite (Kursstartseite), von der aus die übrigen Seiten erreicht werden können. Im Folgenden werden wir die Funktion der für Studierende relevanten Seiten beschreiben. Seiten und Funktionen für Betreuer und Korrektoren sind in der Abbildung nur ausschnittsweise dargestellt. Hierauf gehen wir in „Funktionen für Betreuer Teil I: Betrieb“ aus-

föhrlich ein.



3.2 Die Kursstartseite

Diese Seite ist die zentrale Anlaufstelle für alle Beteiligten. Kursbetreuer sollten Ihren Studierenden die Adresse dieser Seite (oder die der Studentenstartseite) nennen. Eine Prüfung der Zugangsberechtigung findet hier noch nicht statt, sondern erst bei den von hier aus erreichbaren Seiten. Details zur Zugangsberechtigung und den unterschiedlichen Benutzergruppen finden sich in „Zugang zum WebAssign-System“.

Die Kursstartseite kann den Erfordernissen des jeweiligen Kurses angepasst werden. Mehr hierzu findet sich in „Änderbare Seiten“.

3.3 Die Studentenstartseite

Hier sieht man eine nach Aufgabenheften geordnete Übersicht aller schon bearbeiteten bzw. noch zu bearbeitenden Aufgaben. Für Aufgabenhefte, deren Bearbeitungszeitraum noch nicht begonnen hat, sind die einzelnen Aufgaben noch nicht zugänglich.

Über die jeweiligen Aufgaben-Links gelangt man zum eigentlichen Aufgabentext. Schon bearbeitete Aufgaben sind mit einem Stern (*) gekennzeichnet. Durch den Knopf *Heft schließen* gibt man die fertig bearbeiteten Aufgaben zur Korrektur frei. Danach ist in diesem Heft keine Aufgabenbearbeitung mehr möglich (Der Kursbetreuer kann die Hefteinsendung allerdings wieder rückgängig machen).

Wenn man als Betreuer, Korrektor oder Gast auf diese Seite kommt, fehlt der Knopf zum Heft schließen bzw. der Hinweis, dass das Heft geschlossen ist.

Die Studentenstartseite kann den Erfordernissen des jeweiligen Kurses angepasst werden. Mehr hierzu findet sich in „Änderbare Seiten“.

3.4 Aufgabenseite

Hier befindet sich der eigentliche Aufgabentext. Durch Eintragen von Lösungen in die dafür vorgesehen Felder wird die Aufgabe bearbeitet und durch Drücken des Einsendeknopfes (mei-

stens mit *Aufgabe .. einsenden beschriftet*) zum WebAssign-Server abgeschickt. Endgültig abgeschlossen ist die Bearbeitung der Aufgabe jedoch erst, wenn das gesamte Heft geschlossen ist (siehe Studentenstartseite) oder das Bearbeitungsende überschritten ist. Bis dahin kann die Aufgabe beliebig oft eingesandt werden. Zur Unterstützung erscheinen die zuletzt abgeschickten Lösungen in den Eingabefeldern, wenn der Studierende das nächste Mal auf die Aufgaben-seite kommt.

3.5 Quittung

Jede studentische Einsendung wird unmittelbar quittiert. In der Regel werden die Eingaben wiederholt, sowie Angaben zur Aufgabe und der Name des Studierenden ausgegeben. Falls die Aufgabe entsprechend konfiguriert ist, enthält die Quittung auch das Ergebnis einer automatischen Überprüfung der studentischen Eingaben.

3.6 Persönliche Ergebnisse

Hier erhält der Studierende eine Auflistung seiner Korrekturen mit deren Status und ggf. den erreichten Punkten sowie, falls vorhanden und freigegeben, Links zu den einzelnen Musterlösungen.

3.7 Persönliche Kursmappe

Diese Seite ist eine Alternative zur Aufgabenübersicht und der Ergebnisseite. Hier findet man im linken Frame eine Kursmappe. Wenn man auf den Ordner eines Aufgabenheftes klickt und anschließend auf den Ordner einer Aufgabe, kann man sich den Aufgabentext, die korrigierte Einsendung oder die Musterlösung anzeigen lassen bzw. beim Aufgabentext diesen auch bearbeiten. Die ausgewählten Seiten erscheinen jeweils im rechten Frame.

4 Funktionen für Betreuer Teil I: Betrieb

4.1 Die Betreuerstartseite

Diese Seite ist die zentrale Anlaufstelle für Kursbetreuer. Hier können alle administrativen Aufgaben erledigt werden.

Aufgabenverwaltung	Einsendungen und Korrekturen	
Aufgabenübersicht	Einsendungs- und Korrekturübersicht	
Kursressourcen-Upload	Korrekturzipdateiverwaltung Alte Vorgehensweise - Bitte nicht mehr verwenden!!!	
Kursressourcen-Verzeichnis	Erneute Hefteinsendung	
Kursressourcen als Zip-Datei	Statistik	
Korrekturmodul-Verzeichnis		
Studierende	Betreuer	Korrektoren
Belegerliste	Betreuerliste	Korrektorenliste
Studenten anmelden	Betreuer anmelden	Korrektor anmelden
Studenten bearbeiten	Betreuer bearbeiten	Korrektor bearbeiten
Studentenliste		
Nachrichten	Klausuren	Studientage
E-Mail an Kurs-Beleger versenden	Klausuren festlegen	Studientage festlegen

Message of the Day definieren	Klausuranmeldungen	Studentaganmeldungen
	Sonderfall Klausuranmeldungen	
Allgemeines		
Kursparameter definieren		

4.2 Aufgabenübersicht

Hier findet man die Aufgaben nach Aufgabenheften geordnet. Für jede Aufgabe gibt es einen Link, über den man zur Konfiguration der entsprechenden Aufgabe kommt.

Zum Erzeugen neuer Aufgabenhefte gibt man die entsprechende Anzahl in dem ersten Feld unter der Tabelle an und drückt *einfügen*; die neu erzeugten Hefte werden am Ende der Liste angefügt.

Zum Löschen von Aufgabenheften gibt man die entsprechende Anzahl in dem zweiten Feld an und drückt *löschen*; die Hefte werden dann am Ende der Liste entfernt.

Zum Erzeugen von neuen Aufgaben wählt man das Aufgabenheft in dem diese erzeugt werden sollen und gibt in dem vorherigen Feld die Anzahl der zu erzeugenden Aufgaben an und drückt *einfügen*; die Aufgaben werden dann in der Tabelle hinter den schon vorhandenen Aufgaben angefügt.

Zum Löschen von Aufgaben wählt man das Aufgabenheft in dem diese gelöscht werden sollen und gibt in dem vorherigen Feld die Anzahl der zu löschenden Aufgaben an und drückt *löschen*; die Aufgaben werden dann in der Tabelle von unten gelöscht.

Um einem Aufgabenheft einen neuen Bearbeitungsbeginn und ein neues Bearbeitungsende zuzuordnen, gibt man die entsprechenden Daten in die letzten beiden Eingabefelder ein, wählt das zu ändernde Aufgabenheft und drückt anschließend noch den Knopf *ändern*. Falls einem neu angelegtem Aufgabenheft keine Zeit zugeordnet wird, stellt das System automatisch den zu dem vorhergehenden Aufgabenheft folgenden Bearbeitungszeitraum ein.

4.3 Aufgabenkonfiguration

Auf dieser Seite können alle Informationen zu einer einzelnen Aufgabe angesehen und verändert werden. Dazu gehören verschiedene HTML-Seiten sowie Angaben zur Aufgabenstruktur und zur Art der Korrektur bzw. Vorkorrektur. Zu einer Aufgabe gehören vier HTML-Seiten: Aufgabentext, Musterlösung, Hinweise für den Korrektor sowie eine Korrekturseitenschablone. Letztere ist eine HTML-Seite, in die die studentischen Lösungen eingefügt werden und die dann der Korrektor zur Korrektur bekommt. Der Aufgabentext und die Korrekturseitenschablone müssen unbedingt vorhanden sein. Musterlösung und Korrekturhinweise können ggf. fehlen (was wir natürlich nicht empfehlen). Zu diesen vier Seiten kommen je nach Anzahl der Teilaufgaben eine oder mehrere Quittungsschablonen hinzu, das sind Seiten, die der Studierende als sofortige (Online-) Bestätigung seiner Einsendung erhält. Diese Seiten dürfen ebenfalls nicht fehlen.

Bei allen Seiten kann man über den Knopf *generieren* eine einfache Standardversion erzeugen. Diese Generierung liefert nur bei den Korrekturseiten- und Quittungsschablonen unmittelbar verwendbare Ergebnisse, in den anderen Fällen erhält man ein grobes Gerüst, das man als Ausgangsbasis für die Erstellung der eigentlichen Seite verwenden kann. Über die Regeln, die bei der Erstellung von Aufgabenseiten einzuhalten sind, informiert „Funktionen für Betreuer Teil II: Aufgabenseiten“.

Die einzelnen Seiten können bzw. müssen Variablen enthalten, die später, wenn der Studieren-

de oder der Korrektor die Seite zu sehen bekommt, durch aktuelle Werte ersetzt werden. Details hierzu sind im Abschnitt „Variablen“ zu finden. Wenn man sich von dieser Seite aus den Aufgabentext oder eine der anderen HTML-Seiten ansieht, werden die Variablen im „Rohzustand“ angezeigt. Wie der Aufgabentext tatsächlich aussieht, erfährt man, wenn man über die Studentenstartseite oder direkt über den Link *Studenten-Aufgabentext* dorthin kommt. Für alle HTML-Seiten gibt es einen Link auf die jeweilige Seite, der aber erst vorhanden ist, wenn die Seite existiert, sowie einen Knopf zum generieren.

Typisches Vorgehen bei der Neuerstellung des Aufgabentextes:

- Grundgerüst generieren
- Dem Link zum Aufgabentext folgen
- Im Menü *File - Edit Page* bzw. *Datei - Seite bearbeiten* auswählen
- Änderungen vornehmen
- Mit *File - Publish* bzw. *Datei - Seite publizieren* die geänderte Seite zum WebAssign-Server übertragen
- Composer schließen und zur Aufgabenkonfigurierung zurückkehren
- Dem Link *Studenten-Aufgabentext* folgen und das Ergebnis ansehen

Bei späteren Änderungen entfällt natürlich der erste Schritt. Bei den anderen Seiten ist der Ablauf analog.

Des Weiteren finden sich Links zur Studentenstartseite und zum Studenten-Aufgabentext um einen Eindruck davon zu bekommen, wie sich die später zu bearbeitende Aufgabe für den Studenten darstellt.

Typisches Vorgehen zum Überprüfen der „Funktionsfähigkeit“ des Aufgabentextes:

- Aufgabentext ändern (siehe oben)
- Ggf. auf den Knopf *Teilaufgaben erzeugen* drücken
- Ggf. Quittungsschablonen anpassen oder generieren
- Zum *Studenten-Aufgabentext* wechseln
- Eingabefelder ausfüllen
- Auf einen Knopf zum Einsenden drücken – dabei muss man sich als Student identifizieren

Neben den HTML-Seiten gehören zur Konfigurierung einer Aufgabe Angaben zum Aufgabennamen und zur Art der Korrektur bzw. Vorkorrektur. Mit dem Aufgabennamen kann man der Aufgabe eine aussagekräftige Bezeichnung zuordnen, die insbesondere in Übersichten erscheint. Wenn man nichts angibt, heißt die Aufgabe *Aufgabe <n>*, wobei <n> die Nummer der Aufgabe im Aufgabenheft ist.

Details zu Korrektur und Vorkorrektur finden sich in Funktionen für Betreuer Teil III: Korrekturen.

4.4 Der Aufgabenerstellungs-Assistent

Die Erstellung und Konfiguration von zahlreichen Aufgabentypen (Aufgabenseiten und deren automatischer Korrektur) wird durch einen Aufgabenerstellungs-Assistenten unterstützt und erheblich erleichtert. Alle nötigen Informationen werden von dem Assistenten über vordefinierte Masken abgefragt und erfasst. Der Aufgabenerstellungs-Assistent lässt sich über einen Link auf der Seite „Aufgabenerstellung“ in der Aufgabenverwaltung (im oberen Bereich rechts) starten.

Die folgenden Felder definieren die Eigenschaften einer Aufgabenseite:

<i>Seitentitel</i>	Hier sollte unbedingt etwas eingegeben werden, da ansonsten ein Standardwert angenommen wird (Name der Aufgabe).
<i>Einführungstext</i>	Damit ist ein Text gemeint, der gemeinsame Informationen zu allen

	Aufgaben zusammenfasst.
<i>Gesamtbewertung</i>	Es ist möglich, bei der automatischen Korrektur in Abhängigkeit von der erreichten Gesamtpunktzahl bestimmte Kommentare, z.B. „gut“ oder „Sie sollten mehr üben“, auszugeben. Dazu trägt man bei <i>Punkteintervall</i> das gewünschte Intervall, z.B. „0 - 10“, und bei <i>Kommentar</i> den dazugehörenden Kommentar ein.
<i>Korrekturzeitpunkt</i>	Hier kann festgelegt werden, wann die Korrektur erfolgen soll. Bei der Auswahl „ <i>Aufgabe direkt beim Einsenden Korrigieren</i> “ bekommt der Student das Ergebnis der Korrektur sofort nach der Einsendung. Bei der Auswahl „ <i>Aufgabe zum Einsendeschluss Korrigieren</i> “ wird die Korrektur erst beim Einsendeschluss erzeugt und ist dann für den Studenten zugänglich.

Die typische Vorgehensweise zum Erstellen einer Aufgabenseite lässt sich wie folgt zusammenfassen:

- Aufgabenerstellungs-Assistenten aufrufen
- Aufgaben des gewünschten Typs hinzufügen
- die nötigen Felder ausfüllen
- *Aufgabenseite sichern* betätigen

Um eine Aufgabenseite zu erstellen, müssen zunächst Aufgaben hinzugefügt werden. Hierbei hat man die Wahl zwischen verschiedenen Aufgabentypen, deren Besonderheiten im Folgenden noch erörtert werden. Um eine Aufgabe hinzuzufügen, wählt man den gewünschten Typ und drückt auf *Aufgabe hinzufügen*. Man erhält eine Maske, in der zusätzlich Informationen für die Konfiguration der Aufgabe eingegeben werden können.

Wenn man alle Informationen eingegeben hat, muss die Aufgabenseite gesichert werden. Dazu drückt man auf *Aufgabenseite sichern* im unteren Bereich der Seite. Dabei werden die Aufgabenseite, die notwendige Konfiguration für die automatische Korrektur, die Korrekturseite und die Quittungssseite (siehe 3.5) erstellt und gespeichert. Außerdem bekommt man eine Vorschau der fertigen Aufgabenseite. Um noch Änderungen zu machen, kann man auf die vorher angezeigte Seite wechseln und nach den Änderungen wiederholt den Knopf *Aufgabenseite sichern* betätigen.

Derzeit können mit dem Aufgabenerstellungs-Assistenten Aufgabenseiten für die folgenden Aufgabentypen erstellt werden:

- Einfachauswahl,
- Mehrfachauswahl,
- Zuordnung,
- Zahlen,
- Begriffe.

Einige der für Aufgabenseiten festzulegenden Attribute gelten für alle Aufgabentypen. Hierzu gehören die Attribute:

<i>Aufgabentitel</i>	z.B. „Aufgabe 1“
<i>Aufgabentext</i>	Hier soll beschrieben werden, was der Student machen soll.
<i>Punkte</i>	Die maximale Punktzahl, die der Student mit dieser Aufgabe erreichen kann.
<i>Kommentar</i>	Der Kommentar wird in der Lösung angezeigt. Man kann dem Studenten damit zusätzliche Informationen geben, wie z.B. „Diese Aufgabe bezieht sich auf den Stoff von Kapitel XY“.

Darüber hinaus verfügt jeder Aufgabentyp über bestimmte Besonderheiten, die bei der Aufgabenerstellung zu berücksichtigen sind.

Einfachauswahl/Mehrfachauswahl

Bei Auswahlaufgaben haben die Studierenden die Möglichkeit, eine oder mehrere Antworten auszuwählen. Die Eingabemaske enthält dementsprechend Eingabefelder für die möglichen Antworten. Außerdem kann man auswählen, welche Möglichkeiten richtig sind. Hierzu dienen die Auswahlfelder vor jeder möglichen Antwort. Bei Bedarf können alternative Antworten hinzugefügt oder gelöscht werden.

Zuordnung

Die Zuordnung ist eine Erweiterung der Einfachauswahl, wobei die möglichen Antworten zu den Fragen fest vorgegeben sind. Aus der Menge der Antworten sind die korrekten Lösungen durch Markierungen in einer Matrix den entsprechenden Fragen zuzuordnen.

Die Konfiguration erfolgt wie bei den Auswahlaufgaben. Allerdings müssen nun mehrere Fragen eingetragen werden. Die Vorgabe der Lösung erfolgt für jede einzelne Frage durch die korrekte Markierung der Antwort. Die Anzahl der Fragen und Antworten kann variiert werden, d. h. Fragen wie Antworten lassen sich hinzufügen oder löschen.

Zahlen

Die Antwort des Studenten ist bei diesem Aufgabentyp eine Zahl, die in einem bestimmten Intervall liegen muss. Es genügt also eine untere und eine obere Grenze für das Lösungsintervall anzugeben.

Begriffe

Bei einer Begriffsabfrage muss der Studierendende einen oder mehrere Begriffe angeben. Die geforderte Anzahl kann unter *Anzahl der abgefragten Begriffe* eingestellt werden.

Die zu definierende Lösungsmenge kann zu einer Begriffsabfrage mehrere Lösungsworte enthalten (Synonyme). Dies ist nützlich, wenn es mehrere Möglichkeiten oder Synonyme für die gesuchte Lösung gibt oder z. B. gängige deutsche und englische Bezeichnungen existieren. Daher kann zu jedem Begriff eine variierbare Anzahl von Synonymen angegeben werden. Auch die Anzahl der abgefragten Begriffe ist konfigurierbar. Bei der Konfiguration der Musterlösung dürfen nicht benötigte Eingabefelder für Synonyme frei bleiben.

4.5 Kursressourcen-Upload

Unter Kursressourcen versteht man HTML-Seiten und ergänzende Bilder, Filme, Applets usw., die für Aufgaben, Musterlösungen, Quittungs- und Korrektorschablonen sowie Korrekturhinweise verwendet werden. Für die zentralen HTML-Seiten sind feste Namen vorgegeben, z.B. *aufgabe3.1.html* für die erste Aufgabe im dritten Aufgabenheft eines Kurses. Bei diesen Seiten sprechen wir auch von Standard-Kursressourcen. Die Benennung von Kursressourcen muss je Kurs eindeutig sein. Groß- und Kleinschreibung wird unterschieden.

Auf dieser Seite kann man Kursressourcen vom lokalen PC zum WebAssign-Server hochladen. Man kann den Dateinamen direkt eingeben oder über den *Durchsuchen...*- bzw. *Browse ...*-Knopf eine Datei auswählen. Diese wird dann mit dem *Upload*-Knopf in das System hochgeladen.

Derzeit können zip-Dateien sowie ungepackte Einzeldateien hochgeladen werden. Falls erforderlich, werden automatisch neue Aufgabenhefte bzw. Aufgaben erzeugt. Die Struktur der einzelnen Aufgabenseiten wird allerdings nicht analysiert. Dazu muss man in der Aufgabenkonfiguration (siehe Aufgabenkonfiguration) entsprechende Anpassungen manuell

vornehmen.

4.6 Kursressourcen-Verzeichnis

Auf dieser Seite werden alle Kursressourcen aufgelistet – die Standard-Kursressourcen in einer eigenen Liste. Man kann sich die Ressourcen anzeigen lassen und ggf. löschen. Standard-Kursressourcen können allerdings nicht direkt gelöscht werden. Dies geschieht implizit, wenn man Aufgabenhefte, Aufgaben oder Teilaufgaben löscht.

Bei den übrigen Kursressourcen ist das anders. Das WebAssign-System kann derzeit nicht feststellen, zu welcher Aufgaben- oder Musterlösungsseite usw. eine Ressource gehört. Deshalb sollte ein Kursbetreuer nicht mehr benötigte Kursressourcen von Hand löschen (nicht gelöschte Ressourcen richten aber keinen Schaden an).

4.7 Kursressourcen als Zip-Datei

Insbesondere zur Archivierung ist es nützlich, sämtliche Ressourcen eines Kurses aus dem WebAssign-System herauszuholen und auf dem eigenen PC zu sichern. Hier kann man eine entsprechende Zip-Datei erzeugen lassen und herunterladen.

Man sollte diese Funktion aber nicht zum Kopieren eines Kurses von einem Semester zum nächsten verwenden. Dazu gibt es eine effizientere Funktion, die der WebAssign-Administrator auf Anfrage ausführt.

4.8 Korrekturmodul-Verzeichnis

Hier kann man nicht mehr benötigte Korrekturmodule – genauer Klassen – löschen; dazu markiert man ein oder mehrere Klassen in der angezeigten Liste und drückt den Knopf „Selektierte Korrekturmodule löschen“.

4.9 Einsendungs- und Korrekturübersicht

Hier sieht man alle Einsendungen bzw. Korrekturen des aktuellen Kurses im Überblick. Nach Aufgabenheften und Korrektoren geordnet wird angezeigt, wieviele Aufgaben welchen Status (unkorrigiert, beim Korrektor, korrigiert, freigegeben) haben. Je Aufgabenheft und Korrektor führt ein Link zu einer detaillierteren Übersicht der zugehörigen Aufgaben. Im einzelnen können folgende Funktionen ausgeführt werden:

- Die gesamte Hefteinsendung eines Studenten wird über den Knopf *Hefteinsendung löschen* rückgängig gemacht. Dadurch hat der Student wieder die Möglichkeit, (bis zum Bearbeitungsende) an seinen Aufgaben zu arbeiten.
- Wenn es zu einem Aufgabenheft neu eingesendete Aufgabenhefte gibt, die noch nicht an die Korrektoren verteilt sind, kann man das Verteilen hier auslösen. Die dabei angewandte Verteilstrategie wird als globaler Kursparameter festgelegt (siehe Kursparameter definieren). In die Berechnung gehen die aktuell eingestellten Studentenzahlen der Korrektorkräfte ein. Man kann also durch Verändern der Studentenzahlen und ggf. Ändern der Verteilstrategie das Verfahren beeinflussen. Im Nachhinein kann man einzelne Einsendungen einem andere Korrektor zuordnen (siehe Einsendungen und Korrekturen eines Korrektors und eines Aufgabenheftes).
- Sobald eine Einsendung einem Korrektor zugeordnet ist, steht sie diesem als Download zur Verfügung. Dazu muss der Korrektor das automatische Download im Offline-Verfahren anwenden (s. 5.2.).
- (**Alte Vorgehensweise - Bitte nicht mehr verwenden!!!**: Die einem Korrektor zugeordneten Einsendungen zu einem Aufgabenheft kann man zu einer Zip-Datei packen. Diese wird in der Datenbank festgehalten. Dem Korrektor wird automatisch eine E-Mail geschickt, dass

er sich die Datei herunterladen soll.)

4.10 Einsendungen und Korrekturen eines Korrektors und eines Aufgabenheftes

Hier kann der Kursbetreuer die einem Korrektor zugeordneten Einsendungen eines Aufgabenheftes in der Übersicht ansehen und administrieren. Im einzelnen können folgende Funktionen ausgeführt werden:

- Die Aufgaben eines Studenten können über den Knopf *Status zurücksetzen* auf unkorrigiert zurückgesetzt werden. Der Korrektor wird damit zur Überarbeitung seiner Korrektur aufgefordert. Der Betreuer sollte dies dem Korrektor (per E-Mail) mitteilen.
- Die gesamte Hefteinsendung eines Studenten wird über den Knopf *Hefteinsendung löschen* rückgängig gemacht. Dadurch hat der Student wieder die Möglichkeit, (bis zum Bearbeitungsende) an seinen Aufgaben zu arbeiten.
- Die Einsendungen eines Studenten können über den Knopf *Korrektor zuordnen* einem anderem Korrektor zugeteilt werden. Damit lässt sich das Ergebnis der automatischen Verteilung von Einsendungen auf Korrektoren manuell nachkorrigieren.
- Über den Link in der Status-Spalte kann sich der Betreuer den aktuellen Stand der Korrektur ansehen. Beim Status *beim Korrektor* ist dies allerdings nur die letzte im WebAssign-Server aktualisierte Version.

4.11 Typischer Ablauf für die Arbeit eines Betreuers mit Einsendungen

- Nach Einsendung des Studenten werden vom Betreuer die Einsendungen nach einem festen Schema zugeordnet. Diese Zuordnung kann auch manuell verändert werden.
- Der Status der Korrekturen des jeweiligen Aufgabenheftes wird sichtbar, wenn man auf den/die Korrektor(en) klickt, dem dieses Aufgabenheft zugeordnet ist. Der Betreuer kann sich dort auch die Korrekturen selbst, neben den Kommentaren und der Punkte-Vergabe von allen Studenten, die Lösungen eingesandt haben, ansehen.
- Weiterhin können neue Hefteinsendungen vorgenommen werden, auch ohne die bereits eingesandten Hefte zu löschen.
- Es können dort ebenfalls streng selektiv oder auch global Hefteinsendungen gelöscht werden.

4.12 Korrekturzipdateiverwaltung *Alte Vorgehensweise - Bitte nicht mehr verwenden!!!*

Dieses Verfahren soll nur noch angewendet werden, wenn das aktuelle Verfahren der Offline-Korrektur *nicht* funktioniert. Einsendungen bzw. Korrekturen können auch in Form von gepackten Dateien zwischen WebAssign- und lokalem Korrekturserver hin und her bewegt werden. Das Arbeiten mit gepackten Korrekturen und das direkte Herauf- bzw. Herunterladen können nicht gemischt werden. Deshalb müssen sich – zumindest je Aufgabenheft und Korrektor – Kursbetreuer und Korrektor darüber verständigen, welches Verfahren eingesetzt wird.

Auf dieser Seite kann der Kursbetreuer den Status der gepackten Korrekturen ansehen und die vom Korrektor zurückgekommenen Korrekturen in die Datenbank übernehmen.

Typisch für das Arbeiten mit gepackten Korrekturen ist folgender Ablauf:

- Der Kursbetreuer veranlasst das Packen der Korrekturen eines Korrektors (siehe Einsendungs- und Korrekturübersicht).
- Das WebAssign-System schickt dem Korrektor eine E-Mail.
- Der Korrektor lädt die gepackte Datei auf seinen PC herunter.
- Dort packt er sie im zuvor geleerten Verzeichnis *einsendungen* aus (siehe Installation und Start des lokalen Servers).
- Wenn die Korrekturen fertig sind, packt der Korrektor den Inhalt des *einsendungen*-Ver-

- zeichnisses und lädt die gepackte Datei zum WebAssign-Server herauf.
- Das WebAssign-System benachrichtigt den Kursbetreuer, dass fertige Korrekturen angekommen sind.
- Der Kursbetreuer übernimmt die fertigen Korrekturen in die Datenbank und verfährt weiter wie gewohnt, das heißt, er sieht sich die Korrekturen an und gibt sie ggf. frei.

4.13 Erneute Hefteinsendung

Wenn sich herausstellt, dass eine Korrekturseitenschablone fehlerhaft war, aber schon Studenten ihr Aufgabenheft geschlossen haben, kann man hier das Eintragen der studentischen Lösungen in die (überarbeiteten) Korrekturschablonen wiederholen lassen. Eine ggf. konfigurierte automatische Korrektur wird ebenfalls wiederholt. Da in der Regel nur einzelne Aufgaben betroffen sind, wird diese Funktion aufgabenweise durchgeführt. Achtung: Die bis dahin von den Korrektoren vorgenommenen Anmerkungen gehen verloren.

4.14 Statistik

Hier erhält man Auskünfte über die Anzahl der bearbeiteten Aufgaben pro Aufgabenheft und die Anzahl der Studenten, die die Hefteinsendung vorgenommen haben. Angezeigt wird auch, wie oft die Studenten eine Teilaufgabe eingesendet haben, bevor die Hefteinsendung vorgenommen wurde und wie viele Punkte ein Korrektor bei jeder Aufgabe durchschnittlich vergeben hat.

4.15 Belegerliste

Die Belegerliste enthält alle Studenten, die den aktuellen Kurs belegt haben. Man kann die Liste wahlweise nach Namen oder Matrikelnummern sortieren lassen und einzelne Studenten suchen. Hat man den Studenten gefunden, kann man sich detaillierte Informationen anzeigen lassen oder einzelne Angaben ändern, bzw. den Studenten (oder die Belegung) löschen.

4.16 Studenten anmelden

Auf dieser Seite kann man einen Studenten von Hand anmelden. Dazu benötigt man den Namen, die Matrikelnummer und die E-Mail-Adresse des Studenten. Wichtig ist, dass die Matrikelnummer je Veranstalter eindeutig ist, da sie als ID bei der Anmeldung benutzt wird. Das zugehörige Passwort wird automatisch generiert und per E-Mail zugeschickt.

Wenn man einen Studenten anmelden will, der schon einen Kurs des Veranstalters belegt hat, genügt es, dessen Matrikelnummer einzutragen.

4.17 Studenten bearbeiten

Hier wählt man zunächst einen Studenten aus der Belegerliste aus und kann anschließend dessen Eigenschaften ändern. Dazu gehören Name und E-Mail-Adresse sowie die Kurse, die der Student aktuell belegt hat. Achtung: Bei Studenten, die mit ihrer FernUni-Benutzerkennung ($q<Matrikelnummer>$) ins System gekommen sind, ist es nicht sinnvoll, die E-Mail-Adresse zu ändern, da diese immer bei Bedarf vom LDAP-Server gelesen wird. Kursbelegungen können gelöscht werden. Schließlich kann der Student vollständig aus dem WebAssign-System entfernt werden.

4.18 Studentenliste

Hier wird eine Liste mit allen Studenten eines Veranstalters angezeigt, die bislang einen Kurs dieses Veranstalters belegt haben und noch nicht wieder gelöscht worden sind. Die Funktionen auf dieser Seite sind analog zu denen der Belegerliste.

4.19 Betreuerliste

Hier werden sowohl alle beim Veranstalter registrierten Kursbetreuer angezeigt als auch diejenigen, die den aktuellen Kurs betreuen. Man kann einen Betreuer auswählen und sich dazu Details anzeigen lassen und ändern.

4.20 Betreuer anmelden

Hier kann man einen neuen Betreuer registrieren und als Betreuer des aktuellen Kurses eintragen. Man gibt dazu seinen Namen an, der als Schlüssel dient und je Veranstalter eindeutig sein muss, außerdem ein Passwort und eine E-Mail-Adresse. Das Passwort kann der Betreuer später selbst ändern.

Wenn man einen Betreuer anmelden will, der schon als Betreuer des Veranstalters registriert ist, genügt es, dessen Namen einzutragen.

4.21 Betreuer bearbeiten

Analog zur Bearbeitung von Studenteneigenschaften kann man hier Eigenschaften eines Betreuers ansehen und ändern. Insbesondere kann man die Eintragung als Betreuer eines Kurses löschen oder den Betreuer ganz aus dem WebAssign-System entfernen.

4.22 Korrektorenliste

Analog zur Betreuerliste.

4.23 Korrektor anmelden

Analog zur Anmeldung eines Betreuers.

4.24 Korrektor bearbeiten

Analog zur Bearbeitung eines Betreuers. Zusätzlich kann man die Anzahl der Wochenstunden ansehen und ggf. ändern. Die Wochenstundenzahl ist für die Verteilung von Einsendungen wichtig (siehe Einsendungs- und Korrekturübersicht und Kursparameter definieren).

4.25 E-Mail an Kurs-Beleger versenden

Hier kann man eine E-Mail an alle Beleger eines Kurses verschicken. Als Absender wird standardmäßig die Betreuer-E-Mail-Adresse des Kurses eingetragen. Diese kann aber geändert werden. Bei großen Belegerzahlen werden die E-Mails blockweise im Abstand von 15 Minuten verschickt.

4.26 Message of the Day definieren

Im Verlauf eines Kurses gibt es immer wieder wichtige Ankündigungen für die Kursteilnehmer. Damit die Teilnehmer diese Information erhalten, muss sie gut sichtbar platziert werden. Zu diesem Zweck gibt es die Variable \$MOTD, deren Inhalt – ein Text mit oder ohne HTML-Tags – auf verschiedenen Seiten eingeblendet werden kann. Standardmäßig geschieht dies auf der Kurs- und der Studentenstartseite.

Der Inhalt der Variablen \$MOTD kann hier verändert werden. Wenn man auf den Knopf *MOTD speichern* drückt, wird zur Kontrolle der neue Inhalt angezeigt.

4.27 Kursparameter definieren

Hier können verschiedene Eigenschaften gesetzt werden, die global für den ganzen Kurs gültig sind. Derzeit sind dies die folgenden drei Parameter:

Betreuer E-Mail-Adresse: Diese Adresse wird verwendet, wenn das WebAssign-System automatisch eine E-Mail an den Kursbetreuer sendet. Die Adresse ist auch über die Variable *\$BetreuerEmail* verfügbar, womit sie in HTML-Seiten eingesetzt werden kann.

Verteilstrategie: Grundsätzlich richtet sich die Verteilung von Einsendungen auf Korrekturkräfte nach der Anzahl der Wochenstunden. Berücksichtigt wird auch immer nur ein einzelnes Aufgabenheft; ein Ausgleich über mehrere Hefte findet nicht statt. Darüber hinaus sind drei Strategien einstellbar. Die Strategie *freie Zuordnung aufgabenweise* ordnet einzelne Aufgaben zu. Dies ist die Strategie, die die genaueste Annäherung an die Verhältnisse der Wochenstundenzahlen liefert. Die Strategie *freie Zuordnung heftweise* arbeitet analog, lässt allerdings bei jedem Studenten die Aufgaben eines Aufgabenheftes zusammen. Die *feste Zuordnung* schließlich arbeitet analog zur freien heftweisen Zuordnung, wobei aber versucht wird, die Einsendungen eines Studenten dem Korrektor zuzuordnen, der diesen Studenten schon beim vorigen Aufgabenheft korrigiert hat.

Lösungsfreigabe: Musterlösungen können entweder nach Bearbeitungsende oder nach dem Schließen eines Aufgabenheftes freigegeben werden.

5 Funktionen für Betreuer Teil II: Aufgabenseiten

Die Gestaltung der Aufgabentexte soll so wenig wie möglich einschränkt werden. Hier können alle Möglichkeiten von HTML ausgenutzt werden, einschl. Bilder, Tabellen, Frames usw. Die Regeln, die für das WebAssign-System eingehalten werden müssen, betreffen zum einen den Kopf des HTML-Dokuments und zum anderen die Bezeichnung von Eingabefeldern und Knöpfen.

Im Dokumentkopf, genauer in den Angaben zum FORM-Tag, wird festgelegt, um welche Aufgabe es sich handelt und welches Server-Programm die Einsendung entgegennimmt. Durch den Einsatz von Variablen (siehe Variablen) kann der Dokumentkopf für alle Aufgaben gleich gehalten werden, etwa folgendermaßen:

```
<HTML>
  <HEAD>
    <TITLE>Übungen zu "$Kursname"</TITLE>
  </HEAD>
  <BODY>
    <FORM
      METHOD="POST"
      ACTION="$WebAssignServer/$Veranstaltername/Einsendung/$KursNr/$VersionsNr/$AufgabenheftNr/$AufgabenNr"
      NAME="Aufgabe" >

    <H2>Übungen zu "$Kursname"</H2>
    <P><FONT SIZE="+1">$Versionsname</FONT></P>
    <P><FONT SIZE="+1">Aufgabenheft $AufgabenheftNr</FONT></P>
    ...
```

Man kann von diesem Muster abweichen. Wichtig ist aber, dass die Angaben zur FORM exakt stimmen. Die dazugehörigen Zeilen aus dem Muster kann in allen Aufgabenseiten unverändert eingesetzt werden, da sie Variablen verwenden (siehe Variablen). Dadurch erspart man sich Arbeit und eliminiert Fehlerquellen.

5.1 Teilaufgaben

Eine Aufgabe kann in Teilaufgaben zerlegt werden. Hiervon macht man Gebrauch, wenn es möglich sein soll, die Lösung zu Aufgabenteilen getrennt einzusenden, sei es aus technischen Gründen (unterschiedliche Vorkorrektur) oder um der erwarteten Arbeitsweise der Studierenden entgegenzukommen.

Wenn die Aufgabe nicht aus Teilaufgaben besteht, gelten die Angaben zu Teil a) für die ganze Aufgabe.

Zu jeder Teilaufgabe gehört eine Quittungsschablone, die der Studierende bei jedem Einsenden erhält. Die generierte Standard-Quittung dürfte in den meisten Fällen ausreichen. Dabei werden die Eingaben wiederholt sowie Angaben zur Aufgabe und der Name des Studierenden ausgegeben.

Angaben über Anzahl und Struktur (Zahl der Eingabefelder) der Teilaufgaben werden aus dem Aufgabentext extrahiert. Hierzu dienen die Knöpfe *analysieren* und *Teilaufgaben erzeugen*. Beim *Analysieren* wird der Aufgabentext lediglich untersucht und das Ergebnis angezeigt. Beim *Teilaufgaben erzeugen* wird zunächst auch der Aufgabentext analysiert, anschließend werden die Teilaufgaben und deren Eingabefelder in der Datenbank entsprechend angepasst. Dabei werden ggf. Teilaufgaben erzeugt oder gelöscht. Wie man Teilaufgaben und Eingabefelder im Aufgabentext realisiert, findet sich in der Hilfe zum Aufgabentext.

Grundsätzlich reicht eine Form für alle Teilaufgaben. Davon abweichend kann man aber auch für eine einzelne Teilaufgabe eine eigene Form definieren. Dies ist zum Beispiel erforderlich, wenn sich die Teilaufgabe in einem eigenen Frame befindet. Für jede einzelne Form gelten die gleichen Regeln wie für eine Form, die die ganze Aufgabe umfasst (siehe oben). Man beachte aber, dass Forms nicht geschachtelt werden können.

5.2 Eingabefelder

Die Bezeichnung der Eingabefelder ist recht einfach. Unabhängig von ihrem Typ werden sie durchnummeriert und mit *FeldA1*, *FeldA2* usw. bezeichnet. Etwas komplizierter wird es, wenn man mit Teilaufgaben arbeitet. Dann werden die Eingabefelder der ersten Teilaufgabe mit *FeldA1*, *FeldA2* usw. und die Felder der zweiten Teilaufgabe mit *FeldB1*, *FeldB2* usw. bezeichnet. Bei mehr als zwei Teilaufgaben geht es analog weiter. Beispiel:

```
<TEXTAREA NAME="FeldA1" ROWS="8" COLS="75">
  geben Sie hier bitte Ihre Lösung ein
  ...
</TEXTAREA>
```

Bei Eingabefeldern vom Typ CHECKBOX oder RADIO müssen alle Ankreuzmöglichkeiten, die zusammengehören, den gleichen Namen haben – diese zusammengehörigen CHECKBOX- oder RADIO-Inputs werden also als ein Eingabefeld aufgefasst. Beispiel:

```
<INPUT TYPE="checkbox" NAME="FeldA1" VALUE="A">
...
<INPUT TYPE="checkbox" NAME="FeldA1" VALUE="B">
```

In den VALUE-Attributen müssen sich die zusammengehörigen Inputs unterscheiden. Im Allgemeinen hat man hier freie Wahl. Bei Multiple-Choice-Aufgaben der FernUni haben sich A, B, C usw. eingebürgert.

5.3 Knopf zum Einsenden

Ein Knopf zum Einsenden ist vom Typ *submit* und trägt den Namen *einsenden*. Bei mehreren Teilaufgaben braucht man je Teilaufgabe einen solchen Knopf. Diese heißen dann *einsendenA*, *einsendenB*, usw.

```
<INPUT TYPE="submit" NAME="einsenden" VALUE="Aufgaben 1 einsenden">
```

5.4 Applets zur Erweiterung der Eingabemöglichkeiten

Zweifelsohne sind die Eingabemöglichkeiten von HTML-Seiten begrenzt. Es stehen zunächst nur die typischen Formulareingabelemente zur Verfügung, die zur Texteingabe und zum An-

kreuzen bzw. Auswählen aus Listen geeignet sind. Natürlich braucht man, insbesondere bei konstruktiven Aufgaben, weitere Eingabemöglichkeiten, namentlich für mathematische Formeln, Matrizen, Graphiken, formatierte Texte usw.

An dieser Stelle können individuell programmierte Java-Applets für erweiterte Eingabemöglichkeiten sorgen. Wir stellen hier eine Architektur für derartige Applets vor, die in der folgenden Abbildung skizziert wird. Die Applets kommen ohne eigenständige Netzverbindung aus, wodurch zum einen die Architektur sicher und robust wird und zum anderen die Kombinierbarkeit der Applets mit anderen Eingabeelementen gewährleistet ist.

Der im Applet dargestellte bzw. bearbeitete Inhalt (mathematische Formel, Graphik usw.) wird aus einem ggf. unsichtbaren Eingabefeld gelesen und nach der Bearbeitung dort wieder hineingeschrieben (Voraussetzung dazu ist eine Kodierung des Inhalts als String). Der Feldinhalt kann dann zusammen mit den Inhalten anderer Eingabelemente der Seite an den Server übermittelt werden. Die Anfangsbelegung wird analog zu den Anfangswerten der anderen Felder mit der Seite vom Server geliefert.

Technische Voraussetzung für diese Konstruktion ist das Zusammenspiel von Java und JavaScript mit *LifeConnect*, da man vom Java-Applet aus nicht direkt auf die Bestandteile der HTML-Seite, in die das Applet integriert ist, zugreifen kann.

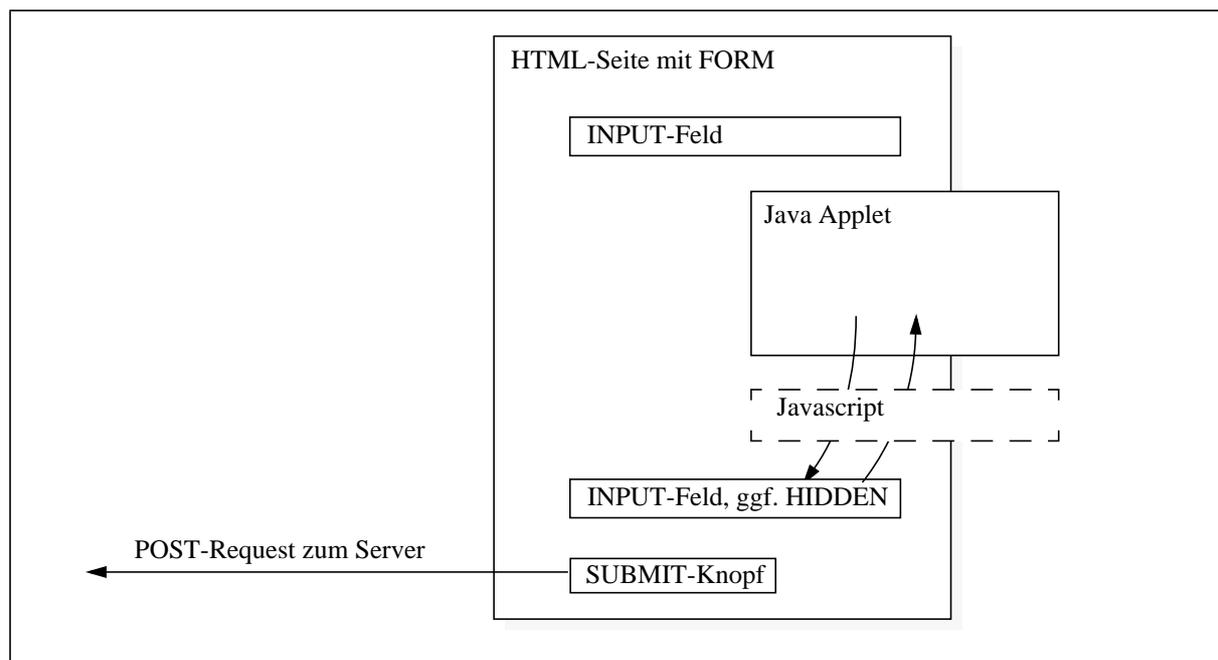


Abb. 2 Integration von Applets in Aufgabenseiten

Leider bringt die Verwendung von LifeConnect eine Beschränkung der einsetzbaren Browser mit sich. Erfolgreich getestet sind Netscape ab Version 4.06 und Internet Explorer ab Version 4.

Wir geben im Folgenden einige Code-Beispiele an, die verschiedene Aspekte der beschriebenen Applet-Architektur illustrieren. Die folgenden beiden Programmstücke zeigen die Verbindung des Applets mit JavaScript, um auf Bestandteile der umliegenden HTML-Seite zugreifen zu können. Voraussetzung ist, dass die Bibliothek *JSObject* importiert wird. Mit *getWindow* und *getMember* kann das Applet durch die Seite navigieren, bis schließlich der Inhalt des gewünschten Eingabefeldes erreicht ist.

```
import netscape.javascript.JSObject;
...
win = JSObject.getWindow (this);
```

```

JSONObject doc = (JSONObject) win.getMember ("document");
JSONObject form = (JSONObject) doc.getMember ("Aufgabe");
JSONObject nameField = (JSONObject) form.getMember ("FeldA1");
String st = (String) nameField.getMember ("value");
ObjectInputStream in = new ObjectInputStream (new ByteArrayInputStream (HexToByteArray
(st)));
graph = (Graph) in.readObject ();
in.close();
graphCanvas.repaint();

```

Gleichzeitig kann man erkennen, wie Java-Objekte über das Netz gesendet werden können. Der gelesene Feldinhalt ist ein String, der zunächst in ein *ByteArray* umgewandelt wird, aus dem dann mit *readObject* ein Java-Objekt, hier ein Graph, einschließlich aller von diesem Objekt erreichbaren Objekte eingelesen wird. Der umgekehrte Weg wird im folgenden Programmstück beschrieben. Dazu dient die Methode *writeObject*. Der schreibende Zugriff auf die umliegende HTML-Seite sieht allerdings etwas anders aus als der lesende. Zunächst wird eine JavaScript-Anweisung als String zusammengebaut. Diese wird dann mit *eval* zur Ausführung gebracht.

```

ByteArrayOutputStream baos = new ByteArrayOutputStream ();
ObjectOutputStream out = new ObjectOutputStream (baos);
out.writeObject(graph);
out.flush();
String st = ByteArrayToHex (baos.toByteArray ());
String command = "document.Aufgabe.FeldA1.value='" + st + "'";
win.eval(command);
out.close();

```

Das dritte Code-Beispiel zeigt, wie das Applet eine Konfigurationsdatei vom Web-Server liest. Der Name der Datei wird über einen Applet-Parameter namens *configuration* festgelegt, der mit der Anweisung *getParameter* gelesen werden kann. Die Methode *getCodeBase* liefert die URL, von der das Applet geladen wurde. Wenn diese URL als Basis verwendet wird, kann das Applet ohne Verletzung von Sicherheitsregeln auf Web-Ressourcen zugreifen.

```

BufferedReader br;
StringBuffer config = new StringBuffer ();
String filename = getParameter ("configuration");
URL codeBase = ((Applet) this).getCodeBase ();
URL url = new URL (codeBase, filename);
br = new BufferedReader (new InputStreamReader (url.openStream ()));
String line = br.readLine ();
while (line != null) {
    config.append (line);
    config.append ("\n");
    line = br.readLine ();
}
br.close();

```

5.5 Einsenden von lokal erstellten Dateien

Auch mit den zuvor beschriebenen Applets stößt man auf Grenzen. Man wird über sie nie die Mächtigkeit eines Standalone-Textprozessors, CAD-Tools oder Formelsatzsystems erreichen. WebAssign bietet daher die Möglichkeit, die Dateiformate externer Werkzeuge (z.B. *.doc, *.pdf, *.tex) zu verarbeiten.

5.5.1 Webbasierter Ansatz vs. externe Dateiformate

Die Verwendung externer Formate bedeutet eine Loslösung vom Ansatz webbasierter Software. Kursbetreuer sollten sich darüber im Klaren sein, dass insbesondere die Verwendung proprietärer Formate (etwa aus dem Hause Microsoft) Probleme mit sich bringen kann:

- fehlende Kompatibilität der Formate verschiedener Produktversionen

- browserseitige Registrierung von Dateierdung bzw. Mimetyp bei allen Benutzern (also insbesondere bei den Studierenden) nötig
- Verteilung und Installation des Tools bei den Studierenden
- Vergrößertes Datenvolumen (hinderlich bei langsamen Internet-Verbindungen)

Es kann aus den genannten Gründen sinnvoll sein, auf den erhöhten Eingabekomfort externer Werkzeuge zu verzichten zugunsten einer rein webbasierten Vorgehensweise.

Im Folgenden stellen wir die Konfigurationselemente vor, die WebAssign für die Verarbeitung von fremdformatigen Einsendungen vorsieht.

5.5.2 Aufgabenseiten

Man bedient sich der folgenden Konstruktion: Über ein Eingabeelement vom Typ *file* kann der Student einen Dateinamen eingeben oder eine Datei auf seinem PC auswählen. Drückt er dann auf den zugehörigen Knopf zum Einsenden, wird der Inhalt der Datei zum WebAssign-Server heraufgeladen.

Für die Erstellung der HTML-Seite ergeben sich einige spezielle Vorgaben: Für das Heraufladen muss man eine eigene *Form* mit einer besonderen Action-URL (geänderter Servicename *EinsendungMultipart*) definieren. Die folgenden Zeilen zeigen dies an einem Beispiel. Man beachte auch den besonderen *ENCTYPE*.

```
<FORM
METHOD="POST"
ENCTYPE="multipart/form-data" ACTION="$WebAssignServer/$Veranstaltername/
EinsendungMultipart/$KursNr/$VersionsNr/$AufgabenheftNr/$AufgabenNr">
...
<INPUT TYPE="file" name="FeldA1">
<INPUT TYPE="submit" NAME="einsendenA" VALUE="Datei einsenden">
</FORM>
```

Man kann sich einen derartigen Rahmen vom System erzeugen lassen, indem man auf der Aufgabenerstellungsseite auf *Generieren* drückt und dabei die Option *mit File-Upload* wählt – dies entspricht der Voreinstellung.

5.5.3 Spezielle Variablen

Es existieren spezielle Variablen, um die Art der Wiedergabe von fremdformatigen Feldinhalten zu beeinflussen. Die Variable `$FeldA1` etwa kann in den folgenden Variationen benutzt werden:

Variable	Bedeutung
<code>\$FeldA1PLAIN</code>	Daten der Einsendung im Klartext
<code>\$FeldA1URL</code>	System-URL der Einsendung
<code>\$FeldA1HREF</code>	System-URL, eingerahmt durch ein <code></code> -Tag (Link)
<code>\$FeldA1IMG</code>	System-URL, eingerahmt durch ein <code></code> -Tag (Anzeige als Bild)

Üblicherweise werden diese Variablen in Quittungs- und Korrektorschablonen benutzt. Merke: in Quittungen entfällt die Angabe der Teilaufgabe, also `$Feld1PLAIN`, `$Feld1URL` usw. .

Sie können jedoch auch in den Aufgabenseiten selbst verwendet werden, um den Studenten über seine letzte Einsendung zu informieren (wie WebAssign das bei anderen Feldtypen standardmäßig vorsieht). In Erweiterung des letzten Beispiels könnte man etwa schreiben:

```

<FORM ...>
  Aufgabe 3:
  (Fragestellung)
  Ihre letzte Einsendung zur Kontrolle: $FeldA1HREF
  <INPUT ... TYPE="FILE" ...>
  <INPUT ... TYPE="SUBMIT" ...>
</FORM>

```

Es bleibt die Frage offen, wie sich das System verhält, wenn man auf den die Darstellung betreffenden Zusatz im Variablennamen (PLAIN, URL, HREF, IMG) verzichtet. Antwort: Es wird je nach Mimetype entschieden, welche Darstellungsform sich am besten eignet:

Mimetype	Darstellungsform
image/jpeg image/gif image/png	IMG
text/* (außer text/ps)	PLAIN
sonst	HREF

Folge daraus ist, dass Standardquittungs- und -korrekturschablonen, so wie sie das System generiert, unverändert für FileUpload-Aufgabenteile verwendet werden können. Im Falle der Korrekturschablonen entscheidet sich das System bei fehlender Angabe grundsätzlich für die Darstellung als Link (HREF). Zusätzlich wird pro Eingabefeld noch ein Upload-Formular für den Korrektor generiert, das – da es in Tags <KORREKTOR> ... </KORREKTOR> eingeschlossen wird – nur für diesen sichtbar ist. Der Korrektor braucht dieses Upload-Formular, da er seinerseits in der Lage sein muss, die korrigierten externen Einsendungen wieder ins System hochzuladen.

Für den Fall, dass der Kursbetreuer nicht mit dem Standardverhalten einverstanden ist (Link auf die Einsendung bzw. letzte Korrektur plus Upload-Formular für den Korrektor), stehen zusätzliche Variablen bereit: \$FELDA1BK {URL/HREF/IMG/PLAIN} verhalten sich analog zu den oben beschriebenen Variablen mit dem Unterschied, dass hier die Korrektur – nicht die Einsendung – referenziert wird. Um das im Fall von \$FeldA1 automatisch generierte Upload-Formular für den Korrektor nachzubilden, steht die Variable \$FeldA1UPLFORM zur Verfügung.

5.6 Konsistenz von Aufgabenseiten, Quittungen und Korrekturschablonen

Nach einer Änderung der Aufgabenseite sollte man deren syntaktische Korrektheit über den Knopf *analysieren* überprüfen lassen. Hat sich die Anzahl der Teilaufgaben und/oder Eingabefelder geändert, muss man über den Knopf *Teilaufgaben erzeugen* dafür sorgen, dass diese Informationen extrahiert und in die Datenbank übernommen werden. Dabei werden evtl. vorhandene Quittungs- und Korrekturseiten nicht geändert (Allerdings werden Quittungsseiten von nicht mehr vorhandenen Teilaufgaben gelöscht). Deren Korrektheit muss von Hand überprüft werden oder man erzeugt über die entsprechenden *generieren*-Knöpfe die Seiten neu.

5.7 Heraufladen von HTML-Seiten

Zum Heraufladen von HTML-Seiten in das WebAssign-System verwendet man den Netscape Composer und dessen Publish-Funktion. Man muss dort eine URL angeben, an die die HTML-Seite und ggf. auch dazugehörige Bilder, Applets usw. geschickt werden sollen. Im Fall der Aufgabenseite hat diese URL die Form:

```

http://www-pi3.fernuni-hagen.de/webassign/<Veranstaltername>/AufgabentextRoh/<KursNr>/<VersionsNr>/aufgabe<AufgabenheftNr>.<AufgabenNr>.html

```

also z.B.

<http://www-pi3.fernuni-hagen.de/webassign/six/AufgabentextRoh/01612/WS98/aufgabe1.3.html>

für den Aufgabentext zu Aufgabe 3 im 1. Aufgabenheft des Kurses 01612 im WS98/99.

Analog werden die URLs für die übrigen Dokumente aufgebaut. Es ändern sich jeweils die (Service-) Bezeichnung hinter dem Veranstalternamen und der Dateiname am Ende der URL. Folgende Übersicht stellt die Angaben für die fünf Dokumentenarten zusammen:

Servicename	Basisname der Seite
AufgabentextRoh	aufgabe
MusterloesungRoh	muster
KorrekturschabloneRoh	korrektur
KorrekturhinweiseRoh	hinweise
QuittungRoh	quittung

Bei der Quittungsschablone kommt die Bezeichnung der Teilaufgabe hinzu. Eine mögliche URL ist z.B.

<http://www-pi3.fernuni-hagen.de/webassign/six/QuittungRoh/01612/WS98/quittung1.3a.html>

für die Quittungsschablone zu Teilaufgabe a) der obigen Aufgabe.

Nützlicher Hinweis: Wenn man vom Browser aus mit *Edit Page* die Seite in den Composer lädt und dort verändert, ist beim Publish die URL automatisch richtig gesetzt. Dadurch erspart man sich Tipparbeit und Fehler.

Eine alternative Möglichkeit zum Heraufladen von Seiten zum WebAssign-Server wird in Her-aufladen von Kursressourcen in Zip-Dateien beschrieben.

5.8 Zum Verständnis der WebAssign-URLs:

Fast alle im WebAssign-System verwendeten URLs sind nach dem folgenden Muster aufge-baut:

[http://www-pi3.fernuni-hagen.de/webassign/<Veranstaltername>/<Servicename>/<KursNr>/<Versi-
onsNr>/...](http://www-pi3.fernuni-hagen.de/webassign/<Veranstaltername>/<Servicename>/<KursNr>/<Versi-
onsNr>/...)

5.9 Bitmaps, Applets und zusätzliche HTML-Seiten

Alle HTML-Seiten, die von den Übungsveranstaltern erstellt oder verändert werden können, können durch zusätzlichen Ressourcen – GIFs, JPEGs, Applets oder zusätzliche HTML-Seiten – ergänzt werden. Die Ergänzungen müssen beim Publish im selben Directory wie die zu publi-zierende Seite liegen; in dieser Seite werden sie als lokale Referenzen eingetragen; im Publish-Dialogfenster gibt man sie in der Liste *include files:* an. Achtung: Die Namen der zusätzlichen Ressourcen müssen je Kursversion eindeutig sein!

5.10 Heraufladen von Kursressourcen in Zip-Dateien

Alternativ zum gerade beschriebenen Verfahren kann man auch mehrere Aufgaben- und andere Seiten mit einem beliebigen HTML-Editor vorbereiten und als gepackte Datei zum WebAssign-Server heraufladen. Das Format der gepackten Datei wird über die Dateiendung identifiziert. Derzeit werden die Endungen (und entsprechenden Formate) *.ZIP* und *.zip* unterstützt. Die Standard-Kursressourcen müssen beim Packen im aktuellen Verzeichnis liegen. Zusätzliche Ressourcen können in Unterverzeichnissen abgelegt sein. Diese Struktur wird dann im WebAs-sign-Server nachgebildet.

5.11 Links auf andere Seiten des WebAssign-Systems

Links auf andere Seiten des WebAssign-Systems können an verschiedenen Stellen nützlich eingesetzt werden, etwa als Verweis auf die nächste Aufgabe oder als Rückverweis auf die Aufgabenübersicht oder die Kursstartseite. Grundsätzlich kann man solche Links sowohl relativ als auch absolut angeben. Darüber hinaus sind für häufig vorkommende Verweise eigene Variablen definiert. Hier einige Beispiele:

Ein Verweis (von der ersten) auf die zweite Aufgabe eines Aufgabenhefts:

absolut:	<code>http://www-pi3.fernuni-hagen.de/webassign/six/Aufgabentext/01612/WS98/1 /2</code>
absolut mit Variablen:	<code>\$WebAssignServer/\$Veranstaltername/Aufgabentext/\$KursNr/\$VersionsNr/1/2</code>
relativ:	<code>../2</code>
spezielle Variable	<code>\$NaechsteAufgabeURL</code>

... auf die erste Aufgabe des nächsten Hefts:

relativ:	<code>../2/1</code>
----------	---------------------

5.12 Links auf Quittungsseiten

Die Situation auf den Quittungsseiten ist etwas anders als auf den Aufgabenseiten: GIFs und andere „Verzierungen“ funktionieren wie gehabt. Da aber die Basis-URL anders ist (statt *Aufgabentext* steht hier *QuittungRoh* bzw. *Einsendung*), funktionieren die relativen Links nicht, wohl aber die speziellen Variablen.

5.13 Probleme mit falschen Links und nicht dargestellten GIFs

Relative Links, also solche zu anderen Seiten des WebAssign-Systems, etwa ein Rückverweis von einer Aufgabenseite auf die Aufgabenübersicht (Studentenstartseite), können vom Netscape Composer zerstört werden. Um das zu verhindern, muss man beim Composer in den *Preferences -> Composer -> Publish* die Option *Maintain links* ausschalten.

Ein ähnliches Problem tritt bei Background Images auf - allerdings werden gerade die nicht-lokalen GIFs falsch behandelt. Hier muss man im Composer unter *Format -> Page Colors and Properties* die Option *Leave image at the original location* anschalten.

5.14 Änderbare Seiten

Neben den HTML-Seiten, die zu den einzelnen Aufgaben gehören - Aufgabentext, Musterlösung, Korrekturhinweise, Korrekturschablone und Quittungen - können auch andere Seiten vom Veranstalter geändert werden. Derzeit trifft dies auf die Kursstartseite, die Studentenstartseite und die Ergebnisübersicht zu.

Die URL für die Studentenstartseite lautet:

```
http://www-pi3.fernuni-hagen.de/webassign/<Veranstaltername>/StudentenStartSeite/<KursNr>/<VersionsNr>
```

Wichtig: Man soll bzw. muss auch auf dieser Seite mit den bekannten Variablen arbeiten (*\$KursNr* usw.). Insbesondere ist die folgende Variable essentiell: Die Tabelle mit der Aufgabenübersicht wird automatisch generiert. Deshalb muss anstelle der Tabelle die Variable *\$Auf-*

gabenliste eingesetzt werden.

Analoges gilt für die anderen änderbaren Seiten. Die folgende Tabelle gibt eine Übersicht über die Servicenamen der entsprechende URLs und die auf den Seiten unbedingt zu verwendenden Variablen:

Servicename der änderbaren Seite	zu benutzenden Variable	Bedeutung der Variablen
KursStartSeite	\$KursStartSeite	Die Links auf die für Betreuer, Korrektoren bzw. Studierende zur Verfügung stehenden Seiten
StudentenStartSeite	\$Aufgabenliste	Die Tabelle mit den Aufgabenheften, Bearbeitungs-terminen und den Links auf die Aufgaben
StudentenKorrekturSeite	\$Ergebnisliste	Die Tabelle mit Informationen über den Zustand einer Korrektur, ggf. dem Link auf die freigegebene Korrektur und die Musterlösung

5.15 Variablen

Mit Hilfe von Variablen kann dafür gesorgt werden, dass in HTML-Dokumente Inhalte (Textstücke) eingefügt werden, die zum Zeitpunkt der Dokumenterstellung noch nicht bekannt sind oder mit Absicht offen gelassen werden. Wichtigste Beispiele sind die Schablonen für Korrekturseiten und Quittungen. Dort sollen die vom Studierenden zuletzt eingegebenen Werte erscheinen. Aber auch an vielen anderen Stellen sind Variablen nützlich und sorgen insbesondere für eine leichte Änderbarkeit, etwa das Verschieben einer Aufgabe in ein anderes Aufgabenheft oder einen anderen Kurs. Im Folgenden geben wir eine Übersicht aller gegenwärtig unterstützten Variablen.

Variablenname	Bedeutung	Gültigkeit
\$Universitaet	Für jeden Veranstalter sind Name der Universität, des Lehrgebiets bzw. des Professors verfügbar.	A, M, H, K, Q
\$Lehrgebiet		
\$Professor		
\$KursNr	Angaben zum Kurs: Zur Unterscheidung des Einsatzes in verschiedenen Semestern dient die VersionsNr, z.B. <i>WS98</i> . Eine ausführliche Form der VersionsNr ist der Versionsname, z.B. <i>Wintersemester 98/99</i> .	A, M, H, K, Q
\$Kursname		
\$VersionsNr		
\$Versionsname		
\$BetreuerEmail	E-Mail Adresse des Kursbetreuers.	A, M, H, K, Q
\$MatrikelNr	Angaben zu einem Studierenden sind nur auf Korrekturseiten und Quittungen zugänglich.	Studentenstartseite, K, Q
\$Nachname		
\$Vorname		

Variablenname	Bedeutung	Gültigkeit
\$Korrektor	Name des Korrektors – nur auf Korrekturseiten zugänglich.	K
\$AufgabenheftNr		A, M, H, K, Q
\$Aufgabenname		A, M, H, K, Q
\$TeilaufgabenNr	Nur in Quittungen verfügbar.	Q
\$FeldA1, \$FeldB1, \$FeldB2, ...	Der Inhalt von Eingabefeldern – was der Studierende in dem entsprechenden Feld zuletzt eingegeben hat.	K
\$Feld1, \$Feld2, ...	In Quittungen sind nur die Felder der zugehörigen Teilaufgabe zugänglich.	Q
\$Feld[A-Z][Nr.]P	Wie FeldA1 etc., speziell für Programmtexte. '<' und '>' wird durch entsprechende HTML-Tags ersetzt.	K
\$NaechsteAufgabeURL	URL für die nächste Aufgabe im selben Aufgabenheft.	A, M, H, K, Q
\$NaechsteAufgabe	Kompletter Link auf die nächste Aufgabe (mit dem Text <i>nächste Aufgabe</i>).	A, M, H, K, Q
\$VorherigeAufgabeURL, \$VorherigeAufgabe	analog.	A, M, H, K, Q
\$KursuebersichtURL, \$Kursuebersicht	URL bzw. Link auf die Kursstartseite.	A, M, H, K, Q
\$AufgabeneuebersichtURL, \$Aufgabeneuebersicht	URL bzw. Link auf die Studentenstartseite.	A, M, H, K, Q
\$AufgabeURL, \$Aufgabe		A, M, H, K, Q
\$WebAssignServer	Bei Verweisen auf andere Seiten des Systems kann diese Variable den Ausdruck <i>http://www-pi3.fenuni-hagen.de/webassign</i> ersetzen.	A, M, H, K, Q
\$MOTD	An der Stelle dieser Variable erscheint im Browser der als <i>Message Of The Day</i> definierte Text.	A, M, H, K, Q
\$Vorkorrektur	Ergebnis der automatischen Vorkorrektur.	Q
\$VorkorrekturA, \$VorkorrekturB, ...	Ergebnis der automatischen Vorkorrektur einer bestimmten Teilaufgabe.	K
\$Vorkorrektur[A-Z]P	Ergebnis der automatischen Vorkorrektur, falls diese Programmtext enthält. (vgl. \$Feld[A-Z][Nr.]P).	K

Variablenname	Bedeutung	Gültigkeit
\$VorkorrekturURLA, \$VorkorrekturURLB, ...	URL eines Bildes, das als Ergebnis der Vorkorrektur oder durch eine Bildeinsendung entstanden ist.	K, Q
\$VorkorrekturIMGA, \$VorkorrekturIMGB, ...	Wir \$VorkorrekturURLA etc., aber eingerahmt von einem HTML-Image-Tag.	K, Q
\$VorkorrekturHREFA, \$VorkorrekturHREFB, ...	Wir \$VorkorrekturURLA etc., aber eingerahmt von einem HTML-HREF-Tag.	K, Q
\$Korrektur	Ergebnis der automatischen Korrektur.	K

6 Funktionen für Betreuer Teil III: Korrekturen

Studentische Einsendungen können zu zwei verschiedenen Zeitpunkten korrigiert bzw. bewertet werden: Zum einen unmittelbar nach der Einsendung, so dass die Bewertung als Bestandteil der Quittung sofort an den Studierenden zurückgeht. Wir sprechen in diesem Fall von Vorkorrektur. Zum anderen können Einsendungen korrigiert werden, nachdem der Studierende durch das Schließen eines Aufgabenheftes seine Einsendungen zu den Aufgaben dieses Heftes für endgültig erklärt hat. Wir sprechen hier von Korrektur oder eigentlicher Korrektur.

6.1 Vorkorrektur

Da die Vorkorrektur bei jeder Einsendung erfolgt, muss teilaufgabenweise festgelegt werden, ob eine Vorkorrektur stattfindet, und wenn ja, welche. Eine Vorkorrektur erfolgt grundsätzlich automatisch. Hierzu dienen sogenannte Korrekturserver; Details hierzu finden sich in Abschnitt 6.5.

6.2 Korrektur

Für die eigentliche Korrektur von Aufgaben gibt es insgesamt vier Möglichkeiten:

- automatisch
- von Hand
- automatisch + von Hand
- keine

Eine automatische Korrektur wird wie bei der Vorkorrektur von einem Korrekturserver übernommen (Abschnitt 6.5). Die Korrektur erfolgt sofort nach dem Schließen des Aufgabenheftes für alle Aufgaben des Heftes. Das Ergebnis wird in die Korrekturseiten eingetragen und diese werden sofort für den Studierenden freigegeben.

Bei der Handkorrektur werden nach dem Schließen des Aufgabenheftes Korrekturseiten mit den studentischen Einsendungen erzeugt. Diese werden von einem Korrektor kommentiert und ggf. bewertet. Nachdem ein Kursbetreuer die Korrektur freigegeben hat, ist sie für den Studierenden zugänglich.

Bei der Kombination von automatischer und Handkorrektur werden die beiden zuvor beschriebenen Abläufe miteinander verbunden: Nach dem Schließen des Heftes läuft die automatische Korrektur durch, das Ergebnis wird in die Korrekturseite eingetragen, die dann der Korrektor zur weiteren Beurteilung erhält.

Die Option *keine Korrektur* ist nur in Kombination mit einer Vorkorrektur sinnvoll, und zwar dann, wenn die Aufgaben automatisch korrigiert werden können und das Speichern von Einsendungen oder Bewertungen nicht erforderlich ist.

6.3 Abläufe und Überprüfung der Korrekturmechanismen

Typischer Ablauf zur Überprüfung der „Funktionsfähigkeit“ einer Korrekturseitenschablone (für Handkorrektur):

- Korrekturseitenschablone erstellen oder generieren (nach dem Generieren gibt es eigentlich nichts zu Überprüfen).
- Als Student Einsendungen zu allen Teilaufgaben einer Aufgabe vornehmen.
- Auf der Studentenstartseite den Knopf *Heft schließen* drücken.
- Als Betreuer zur Einsendung- und Korrekturverwaltung gehen.
- Die neuen Einsendungen des betreffenden Aufgabenhefts verteilen und dabei darauf achten, welcher Korrektor die Einsendung bekommt.
- Zu den Korrekturen diese Korrektors gehen und die neue Einsendung ansehen.
- Um den Vorgang wiederholen zu können, muss das Schließen des Heftes rückgängig gemacht werden. Dazu dient der Knopf 'Hefteinsendung löschen', womit alle Korrekturen eines Studenten für ein Aufgabenheft gelöscht werden.

Typischer Ablauf zur Installation und Überprüfung einer Vorkorrektur:

- Unter *Vorkorrekturmodul* den Namen des Korrekturserver, der die Vorkorrektur durchführen soll.
- Sicherstellen, dass der ausgewählte Korrekturserver verfügbar ist (entfällt, falls es sich um einen vordefinierten Server handelt)
- Als Student eine Einsendung zur betreffenden Teilaufgabe vornehmen.
- Die Vorkorrektur erscheint auf der Quittung.

Typischer Ablauf zur Installation und Überprüfung einer automatischen Korrektur:

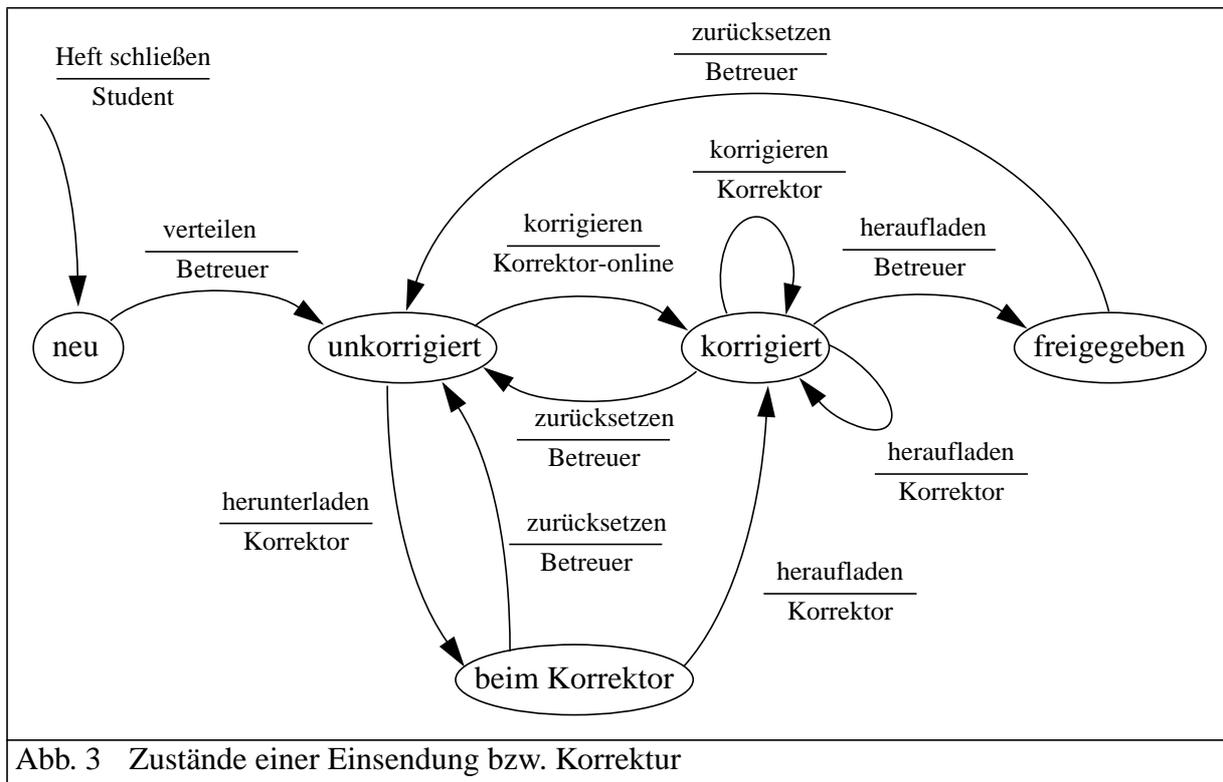
- Die Korrekturart auf *automatisch* stellen.
- In der Korrekturseitenschablone an geeigneter Stelle die Variable $\$Korrektur$ einfügen.
- Unter *Korrekturmodul* den Namen des Korrekturserver eintragen, die die Korrektur durchführen soll.
- Analog zur Vorkorrektur sicherstellen, dass der Korrekturserver verfügbar .
- Als Student Einsendungen zu allen Teilaufgaben einer Aufgabe vornehmen.
- Auf der Studentenstartseite den Knopf *Heft schließen* drücken.
- Zu den persönlichen Ergebnissen wechseln und die Korrektur ansehen.
- Um den Vorgang wiederholen zu können, muss wieder das Schließen des Heftes rückgängig gemacht werden.

6.4 Der Lebenszyklus von Handkorrekturen

Korrekturen entstehen, wenn ein Student ein Aufgabenheft schließt. Dann wird für jede Aufgabe des Heftes aus den studentischen Einsendungen und der Korrekturseitenschablone ein Korrektur-Objekt zusammengesetzt. Wenn die zugehörige Aufgabe von Hand korrigiert wird, durchläuft das Korrektur-Objekt nun verschiedene Stadien. Zunächst hat es den Zustand *neu*. Dieser ändert sich zu *unkorrigiert*, wenn der Betreuer sie einem Korrektor zugeordnet hat (über den Knopf *Neue Einsendungen verteilen*). Anschließend kann der zugeordnete Korrektor die Korrektur bearbeiten. Sie bekommt dabei den Zustand *korrigiert*. Schließlich setzt der Betreuer den Zustand auf *freigegeben*. Erst in diesem Zustand kann der Student sich das Ergebnis ansehen.

Veränderungen am Inhalt der Korrektur selbst und an der dazugehörigen Punktezahl kann also nur der Korrektor vornehmen, dem die Korrektur zugeordnet worden ist. Ein Betreuer kann eingreifen, indem er die Korrektur am Ende freigibt oder, wenn er nicht zufrieden ist, den Zustand auf *unkorrigiert* zurücksetzt.

Neben dem gerade beschriebenen Ablauf ist auch eine offline Korrektur möglich. Das Diagramm in Abb. 3 zeigt alle möglichen Zustandsübergänge.



Wenn ein Korrektor dieselbe Korrektur sowohl online als auch offline bearbeitet, können Konflikte auftreten. Das WebAssign-System ist hier wenig restriktiv – der Korrektor muss selbst darauf achten, dass er seine eigene Arbeit nicht überschreibt. Wie aus dem Diagramm zu entnehmen ist, kann eine Korrektur, die heruntergeladen ist, auch online noch korrigiert werden. Lädt man die offline bearbeitete Korrektur später doch noch herauf, überschreibt diese die zuvor online bearbeitete Version.

6.5 Automatische Korrektur über Korrekturserver

Für die automatische Korrektur und Vorkorrektur bietet WebAssign das Konzept sogenannter *Korrekturserver*, abgekürzt *KServer*, an. Korrekturserver sind eigenständige Software-Module, die als vom Kernsystem losgelöste Server betrieben werden und Requests über CORBA annehmen. Die Wahl von CORBA als Kommunikationsgrundlage gewährleistet bei der Realisierung eines KServers die freie Wahl der Programmiersprache – vorausgesetzt es gibt eine CORBA-Implementierung für sie.

Prinzipiell funktioniert die Kommunikation so: beim Start des KServers meldet sich dieser beim Kern-WebAssign-System mit seinem eindeutigen Namen und seiner IOR (Interoperable Object Reference) an und wird registriert. Der so bekannt gemachte Name kann bei der Aufgabenkonfiguration als Korrektur- oder Vorkorrekturmodul verwendet werden. Wird im laufenden Betrieb die automatische Korrektur vom WebAssign-System angestoßen (dies geschieht in der Regel zum Zeitpunkt der Hefteinsendung), setzt das Kernsystem für jede von einem Studenten eingesandte Aufgabe oder Teilaufgabe einen Request an den KServer ab. Der KServer korrigiert die Aufgabe und liefert das Ergebnis zurück. Das Kernsystem bindet schließlich das Resultat in die Korrekturseite ein.

Bei den Korrekturservern wird unterschieden zwischen kursspezifischen, vom jeweiligen Ver-

anstanter betriebenen Servern und vordefinierten Servern, die das WebAssign-Kernsystem intern startet.

6.5.1 Kursspezifische KServer

Voraussetzung für den Betrieb eines kursspezifischen KServers ist die Vereinbarkeit mit den lokalen Sicherheitsvorkehrungen. Genauer gesagt müssen die CORBA-spezifischen Ports für die Kommunikation mit dem WebAssignServer offen gehalten werden. Voraussetzung für die Entwicklung ist die Kenntnis der WebAssign-IDL, welche in der technischen Dokumentation zu finden ist, und die Verfügbarkeit einer CORBA-Implementation für die verwendete Programmiersprache.

Die am Lehrgebiet Praktische Informatik III der FernUniversität Hagen (<http://www.informatik.fernuni-hagen.de/import/pi3/>) betriebenen KServer sind in Java realisiert. Es wird eine an der FU Berlin entwickelte CORBA-Implementation namens JACORB eingesetzt (<http://jacorb.inf.fu-berlin.de/>). Über den in JACORB enthaltenen IDL-Parser generieren wir aus der WebAssign-IDL zugehörige Java-Klassen (Paket *WebAssign.corba*). Weiter leiten wir KServer-Klassen von einer abstrakten Oberklasse namens *AbstractKorrekturserver* ab. Diese Oberklasse implementiert die Funktionalität, die die reine Kommunikation betrifft, etwa den weiter oben beschriebenen Registrierungsmechanismus. Im Folgenden stellen wir die Klasse *DemoKorrekturserver* vor, die am Lehrgebiet als Referenz bei der Entwicklung von ESevern verwendet wird.

```
package WebAssign.korrektur.demo;

import org.omg.CORBA.*;

import WebAssign.corba.ServerKorrektur;
import WebAssign.corba.WebAssignStudent;
import WebAssign.corba.StudentenEinsendung;
import WebAssign.korrektur.utils.AbstractKorrekturserver;
/**
 * Diese Klasse demonstriert den Aufbau von Korrekturservern.
 */
public class DemoKorrekturserver extends AbstractKorrekturserver{

    /**
     * Diese Methode wird bei Einsendung einer Teilaufgabe aufgerufen.
     * Sie kann die Daten verarbeiten
     * und liefert als Ergebnis die Korrektur der Einsendung zurueck
     * @param student - der Student, der seine Aufgabe eingesandt hat
     * @param eigenschaften - ein Text, der die Konfiguration fuer die Korrektur enthaelt
     * @param einsendungen - die eingesendeten Daten
     * @return die Korrektur der Einsendung
     */
    public ServerKorrektur[] teilaufgabeEinsenden(
        WebAssignStudent student,
        java.lang.String eigenschaften,
        StudentenEinsendung[] einsendungen) {

        /* je nachdem, ob bei der Korrektur gemeinsame Ressourcen verwendet werden
         * (z.B. einen Compiler, der seine Ausgaben immer in die selbe Datei schreibt),
         * sollte der Ablauf synchronisiert werden z.B. mit:
         */
        //Hier kann man nach Belieben die Korrektur vornehmen

        StringBuffer ergebnis = new StringBuffer();
        //Die Matrikelnummer des Studenten in den Ergebnistext schreiben
        ergebnis.append("Request von " + student.matrikelnr);

        //Die Feldinhalte auslesen und in den Ergebnistext schreiben:
        for (int i = 0; i<einsendungen.length; i++) {
```

```

        StudentenEinsendung ein = einsendungen[i];
        String p = new String(ein.einsendung);
        ergebnis.append("Eingabe in Feld " + (i+1) + ":" + p);
    }

    //Serverkorrektur erzeugen
    WebAssign.corba.ServerKorrektur[] sk = new WebAssign.corba.ServerKorrektur[1];
    String name = "";

    //Eintrag fuer die Serverkorrektur erzeugen
    WebAssign.corba.ServerKorrektur k = new WebAssign.corba.ServerKorrektur(name,
        ergebnis.toString().getBytes());
    //Eintrag in die Serverkorrektur einfüegen
    sk[0] = k;

    return sk;
}

/**
 * Diese Methode wird bei Einsendung einer Aufgabe aufgerufen. Sie kann die Daten
 * verarbeiten
 * und liefert als Ergebnis die Korrektur der Einsendung zurueck
 * @param student - der Student, der seine Aufgabe eingesandt hat
 * @param eigenschaften - ein Text, der die Konfiguration fuer die Korrektur enthält
 * @param einsendungen - ein Array, dass fuer jede Teilaufgabe die eingesendeten Daten
 * enthaelt
 * @param ih - dieser Variablen werden am Ende der Korrektur die erreichten Punkte zug
 * wiesen
 * @return die Korrektur der Einsendung
 */
public WebAssign.corba.ServerKorrektur[] aufgabeEinsenden(
    WebAssign.corba.WebAssignStudent student,
    java.lang.String eigenschaften,
    WebAssign.corba.StudentenEinsendung[][] ae,
    org.omg.CORBA.IntHolder ih) {
    //hier kann man genauso vorgehen wie in teilaufgabeEinsenden

    int punkte = 0;

    WebAssign.corba.ServerKorrektur [] sk = null;
    //die Punktzahl muss zugewiesen werden, damit sie im WebAssign gespeichert werden kann
    ih.value = punkte;
    return sk;
}

/**
 * Liefert den Namen des Korrekturserver zurueck
 * Diese Methode muss in Unterklassen implementiert werden.
 */
public String getKServerName(){
    return KServerNAME;
}

//Der Name des Korrekturserver
final String KServerNAME = "DemoKorrekturserver";

/**
 * Diese Methode dient dazu, den Korrekturserver von der Kommandozeile aus zu starten
 * @param args - die Kommandozeilenparameter,
 */
public static void main(String[] args){
    if (args.length > 0){
        DemoKorrekturserver aes = new DemoKorrekturserver();
        connect(aes, args[0]);
    }else System.out.println("Starten mit:\n java DemoKorrekturserver
        konfigurationsdatei");
}

```

```
}  
}
```

6.5.2 Vordefinierte KServer

Zur Bewertung von Multiple-Choice-Aufgaben und einfachen numerischen Aufgaben können vordefinierte KServer eingesetzt werden, sowohl zur Vorkorrektur als auch für die eigentliche Korrektur. Benutzt man zur Erstellung dieser Aufgaben den Aufgabenerstellungsassistent (siehe Abschnitt 4.4), ist die Korrektur über die vorgefertigten Korrekturserver automatisch richtig konfiguriert.

Es kann jedoch auch vorkommen, dass man die Konfiguration von Hand bearbeiten bzw. nachbearbeiten will. Wir geben daher im Folgenden exemplarisch den typischen Arbeitsablauf bei der Konfiguration einer Multiple-Choice-Vorkorrektur an:

- Art der Korrektur auf *keine* stellen, eine Korrekturschablone wird nicht benötigt
- beim Vorkorrekturmodul das Wort *TeilaufgabenBewerter* eintragen
- auf *Änderungen sichern* drücken
- auf *Eigenschaften ...* drücken – dann geht ein Fenster auf
- in diesem Fenster einen Konfigurationstext der u.a. Form eintragen
- unten im Fenster auf den Knopf *fertig* drücken – das Fenster wird geschlossen
- auf *Änderungen sichern* drücken
- die Quittung ändern (ggf. erst generieren): Dort soll statt *In Feld 1 haben Sie eingegeben: \$Feld1* nur noch *\$Vorkorrektur* stehen.

Beispiel für eine einfache Quittung:

```
Bestätigung Ihrer Einsendung  
  
von $Vorname $Nachname  
  
$Vorkorrektur
```

Beispiel für einen Konfigurationstext:

```
Schablonenoption:  
Musterloesung  
Gesamtbewertung  
  
Init:  
Aufgabentyp: XausN  
Vorgabe: A,B,C,D; B,D  
Punkte: 10  
Gewichtung: 1  
N: 4  
Name:Aufgabe 15  
Kommentar: Kommentar1  
  
Init:  
Aufgabentyp: lausN  
Vorgabe: D  
Punkte:10  
Gewichtung:1  
N:4  
Name: Aufgabe 16  
Kommentar:  
  
Init:  
Aufgabentyp: Zahlen  
Vorgabe: 1.1 - 1.2  
Punkte:10  
Gewichtung:1  
N:1  
Name: Aufgabe 17
```

Intervall:0 10
Textbaustein:Sie sollten viel mehr ueben!

Intervall:11 20
Textbaustein:Sie sollten mehr ueben!

Intervall:21 30
Textbaustein:Sie haben den Stoff verstanden!

Erläuterungen zum Konfigurationstext:

Bei *Schablonenoption*: kann man einstellen, ob die richtige Lösung (Musterlösung) bzw. die Gesamtbewertung angezeigt wird. Diesen Teil kann man weglassen. Dann werden sowohl Musterlösung als auch Gesamtbewertung angezeigt.

Für jede Frage (Eingabefeld) gibt man hinter dem Schlüsselwort *Init*: an, wie das Feld korrigiert werden soll. Bei mehreren Feldern müssen diese in der gleichen Reihenfolge wie im Aufgabentext beschrieben werden. Hinter *Init*: können die folgenden Angaben gemacht werden.

- Hinter *Aufgabentyp*: steht *IausN*, *XausN* oder *Zahlen*. Diese Angabe ist obligatorisch.
- Hinter *N*: steht die Anzahl der Ankreuzmöglichkeiten bei *XausN*-Feldern.
- Hinter *Vorgabe*: steht die richtige Lösung, bzw. mehrere richtige Lösungen durch ein Semikolon getrennt. Bei *Zahlen*-Feldern müssen hier eine untere und eine obere Grenze für die zu prüfende Zahl stehen. Diese Angabe ist obligatorisch.
- Hinter *Name*: kann man einen Namen angeben, der das Feld in der Auswertung bezeichnet. Dieser sollte mit einer entsprechenden Bezeichnung im Aufgabentext übereinstimmen.
- Hinter *Kommentar*: kann man einen Text angeben, der bei der Einzelbewertung angezeigt wird.
- Hinter *Punkte*: und *Gewichtung*: gibt man an, wieviele Punkte maximal für das Feld vergeben werden.

Wenn man je nach erreichter Punktzahl unterschiedliche Texte bei der Gesamtbewertung will, muss man dies mit *Intervall*: und *Textbaustein*: spezifizieren.

7 Funktionen für Korrektoren

Korrektoren können wahlweise online oder offline arbeiten. Für die offline-Korrektur muss zunächst ein entsprechender Server auf dem lokalen PC installiert werden (siehe Installation und Start des lokalen Servers).

7.1 Online-Korrektur

7.1.1 Korrektorstartseite

Hier sieht ein Korrektor die ihm zugeordneten Aufgaben im Überblick. Nach Aufgabenheften geordnet wird angezeigt, wie viele Aufgaben welchen Status (unkorrigiert, beim Korrektor, korrigiert, freigegeben) haben. Je Aufgabenheft führt ein Link zu einer detaillierteren Übersicht der zugehörigen Aufgaben.

7.1.2 Einsendungen und Korrekturen eines Aufgabenhefts

Die Einsendungen zu einem Aufgabenheft sind hier tabellarisch angeordnet. In einer Zeile sind die Einsendungen eines Studenten, in einer Spalte die Einsendungen zu einer Aufgabe des Heftes angeordnet. Je Einsendung werden Status und aktueller Punktestand angezeigt. Ein Link führt zur Korrektur der einzelnen Einsendung.

Außerdem finden sich auf dieser Seite Funktionen für das Herunter- und Heraufladen von ge-

packten Korrekturen. Diese Funktionen, manifest in zwei Buttons, betreffen *ausschließlich die veraltete Vorgehensweise*. Das erfolgreiche Herunterladen muss man bestätigen, indem der Zustand der gepackten Korrekturen im WebAssign-Server auf *beim Korrektor* setzt. Zum Herauf-laden wird die gepackte Datei ausgewählt und der Knopf *Upload Korrekturen für Heft <n>* gedrückt.

7.1.3 Korrektur einzelner Aufgaben

Hier kann man die Einsendung zu einer einzelnen Aufgabe korrigieren und die zugehörige Punktzahl verändern. Über den Link *Hinweise zur Korrektur* öffnet sich ein Fenster, das Hinweise oder Korrekturrichtlinien des Betreuers enthält.

Zum Korrigieren eine Aufgabe geht ein Korrektor folgendermaßen vor:

- Mit der linken Maustaste wird einmal in den Frame mit der Aufgabe geklickt.
- Den Menüpunkt *File - Edit Frame* bzw. *Datei - Rahmen Bearbeiten* auswählen.
- Änderungen vornehmen (die Aufgabe korrigieren).
- Den Menüpunkt *File - Publish* bzw. *Datei - Seite Publizieren* auswählen.
- Das Composer-Fenster schließen.

Durch Reload wird die geänderte Korrektur sichtbar. Der Zustand der Korrektur ist nun auf *korrigiert* gesetzt.

Zum Ändern der Punktzahl führt man folgende Schritte aus:

- Die Punktzahl in das Feld *Punkte neu* eintragen.
- Mit dem Knopf *Änderungen speichern* wird die Punktzahl gesichert und der Status der Aufgabe auf korrigiert gesetzt.

7.2 Offline-Korrektur

Mit dem Korrekturkit für offline Korrekturen wird eine gebührensparende Arbeitsweise unterstützt: Ein Korrektor kann von seinem lokalen PC aus alle anliegenden Korrekturen mit einem Knopfdruck auf seinen PC herunterladen, diese dann offline bearbeiten und die fertigen Korrekturen schließlich mit einem Knopfdruck wieder auf den WebAssign-Server hochladen.

Zu diesem Zweck muss auf dem lokalen PC zunächst ein Korrekturkit installiert werden. Dabei handelt es sich um einen kleinen WWW-Server, der unter <http://127.0.0.1:8189> zu erreichen ist.

7.2.1 Korrektorstartseite

Zur Offline Korrektur hat der Korrektor zunächst die dazu benötigte Software auf seinen PC zu laden. Diese Software besteht aus zwei Elementen:

Das *Offline- Korrekturkit* findet er auf der CD *Software für Korrektoren*, die ihm die FernUniversität zugesandt hat. Den auf dieser CD festgehaltenen Anweisungen folgend, installiert der Korrektor das Offline- Korrekturkit. Wahlweise kann der Korrektor dieses Offline- Korrekturkit auch unter dem Punkt *Software für Korrektoren* auf der jeweiligen Kursstartseite herunterladen. Ebenfalls auf der CD befinden sich das benötigte Java Runtime Environment und eine neue Version des ebenfalls benötigten Netscape-Browsers.

Die *Korrektur- Initialisierungs- Datei* (ini-Datei) findet er auf der Kursstartseite des jeweiligen Kurses unter *Software für Korrektoren*. Ferner findet er dort auch die Möglichkeit, das Offline-Korrekturkit und das benötigte Java Runtime Environment auf seinen PC herunterzuladen. Dazu muss er während der Arbeitssitzung den jeweiligen Dialog-Anweisungen folgen.

7.2.2 Installation und Start des lokalen Servers

Der lokale Server ist als Java-Programm realisiert. Deshalb muss zum Starten eine Java-Laufzeitumgebung vorhanden sein. Das kann JDK sein (ab Version 1.1.5) oder die reine Laufzeitumgebung JRE. Letztere findet man unter der Adresse <http://java.sun.com/products/jdk/1.1/jre/> oder auf der Installations-CD.

Die Server-Software muss vom WebAssign-Server heruntergeladen (auf der Kursstartseite den Link *Software für Korrektoren* verfolgen) werden oder von der Installations-CD kopiert werden. Dies geschieht durch Ausführen von *install.vbs*. Hiermit wird nicht nur die erforderliche Serversoftware heruntergeladen und installiert, sondern gleichzeitig auch das Standardverzeichnis C:\Programme\Korrektorkit angelegt. In dieses Verzeichnis muss nun noch die eine Initialisierungsdatei des/der jeweiligen Kurse(s) von der Webseite des entsprechenden Kurses heruntergeladen werden. Bei der Ausführung von *install.vbs* wird ebenfalls noch angeboten, die Java-Laufzeitumgebung herunterzuladen sowie eine neue Version des Netscape-Browsers. Beides ist auch getrennt voneinander möglich.

Es wird nun automatisch eine Verknüpfung erstellt und auf den Desktop des PCs ein Logo „KServ“ *StartKor.kit* angelegt. Gestartet wird das *Korrektorkit* durch einen Doppelklick auf das „KServ“-Logo - es öffnet sich ein Netscape-Fenster.

7.2.3 Einstellungen der Software für Korrektoren

Der Korrektor kann die Übertragungsparameter zum WebAssign-Server durch Änderung von vier Werten beeinflussen. Diese Parameter findet er unter *Datenaustausch -Einstellungen* auf der Startseite des *Offline-Korrekturkits*:

- **Paketgröße:**
Dieser Parameter bestimmt die Anzahl der Korrekturen, die in einem Schritt vom WebAssign-Server heruntergeladen werden. Voreingestellt sind 30 Korrekturen. Wenn der Korrektor über eine stabile Internetverbindung verfügt, kann diese Anzahl beibehalten. Sollte die Verbindung häufiger zusammenbrechen, ist die Anzahl der herunterzuladenden Korrekturen zu verringern.
- **Verbindungsversuche:**
Hier wird die Anzahl der Versuche angegeben, den WebAssign-Server zu erreichen, die je Transaktionsvorgang durchgeführt werden.
- **Protokoll-Refresh-Rate:**
Im rechten Frame der Seite *Datenaustausch* im *Offline-Korrekturkit* erscheint bei laufendem Transfer ein Protokoll. Dieser Parameter gibt an, in welchem Sekundenabstand sich das Protokoll fortschreiben soll.
Dieses Protokoll lässt sich im linken Frame der Seite *Datenaustausch* manuell löschen. Dort kann man den laufenden Transfer auch manuell unterbrechen.
- **Timeout beim Verbindungsaufbau:**
Welche Zeit maximal vergehen soll, bis ein Versuch eines Verbindungsaufbaus zum WebAssign-Server automatisch abgebrochen werden soll, lässt sich mit diesem Parameter in Sekunden einstellen.

7.2.4 Herunter- und Heraufladen von gepackten Korrekturen

Als robuste Variante zum gerade beschriebenen Verfahren können Korrekturen auch in Form von gepackten Dateien zwischen WebAssign- und lokalem Server hin und her bewegt werden. Dazu muss ein Kursbetreuer zunächst das Packen der Korrekturen eines Korrektors veranlassen. Der Korrektor wird mit einer automatisch verschickten E-Mail davon in Kenntnis gesetzt. Er kann nun die gepackte Datei herunterladen. Auf seinem Rechner muss er sie im zuvor ge-

leerten Verzeichnis *einsendungen* (siehe oben) auspacken. Wenn die Korrekturen fertig sind, packt der Korrektor den Inhalt des *einsendungen*-Verzeichnisses und lädt die gepackte Datei zum WebAssign-Server herauf.

7.2.5 Einsendungen und Korrekturen eines Aufgabenhefts

Ist das Offline-Korrekturkit auf dem PC des Korrektors gestartet, öffnet sich ein Netscape Browser-Fenster. Dann nimmt der PC Verbindung zum Korrektur-Server auf. Ist diese hergestellt, erscheint auf dem PC-Bildschirm des Korrektors die Übersicht für den Korrektor eines seiner Kurse. Direkt über der Überschrift befindet sich ein Pulldown-Fenster, dass es dem Korrektor erlaubt, zwischen allen Offline-Korrekturstartseiten der ihm zugeordneten Kurse zu wechseln. Dadurch ist es nicht nötig, bei einem Wechsel des bearbeiteten Kurses, die entsprechende Kurs-Konfiguration mit der ihr zugehörigen Korrektur-Initialisierungsdatei zu starten.

7.2.6 Zustände bei der offline Korrektur

Analog zur online Korrektur zeigt das Diagramm in Abb. 4 die möglichen Zustände einer auf den lokalen PC heruntergeladenen Korrektur. Alle Zustandsübergänge werden durch Aktionen des (auf dem lokalen Server arbeitenden) Korrektors ausgelöst. Man beachte, dass beim Heraufladen nur die Korrekturen im Zustand *korrigiert* berücksichtigt werden.

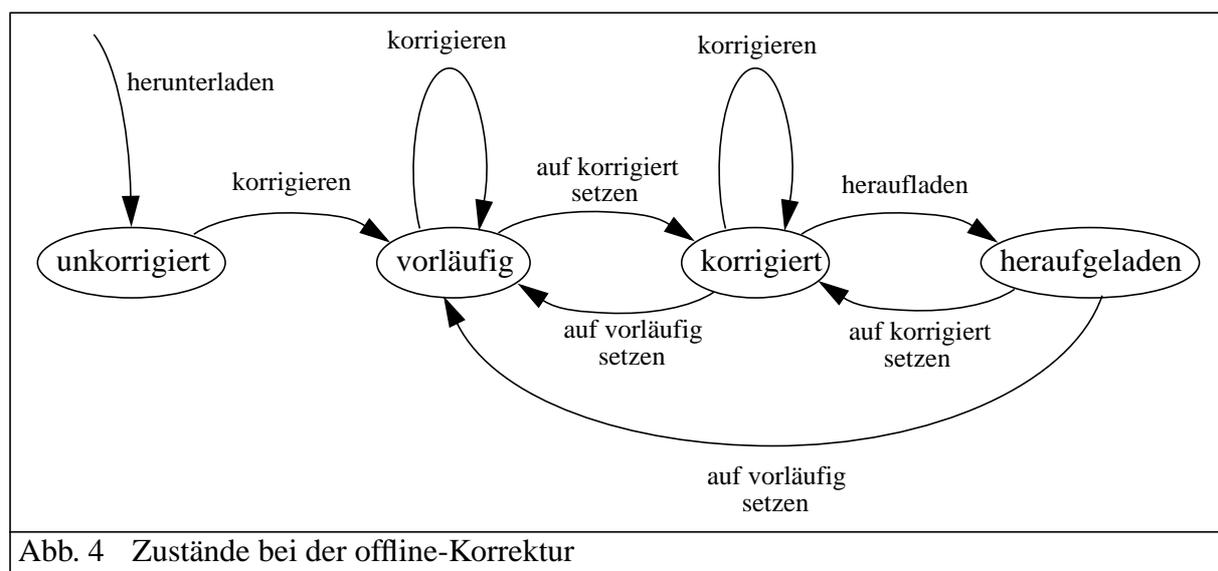


Abb. 4 Zustände bei der offline-Korrektur

7.2.7 Korrektur einzelner Aufgaben

Analog zur Online-Version (s.5.1.3.). Allerdings fehlt hier der Link auf die Korrekturhinweise. Zusätzlich gibt es Navigationshilfen zur gleichen Aufgabe des nächsten bzw. vorherigen Studenten in der Liste der heruntergeladenen Einsendungen.

Im Unterschied zur online Version kann der Status einer Korrektur hier manuell gesetzt werden. Damit ist insbesondere eine Unterscheidung zwischen den Zuständen *vorläufig* und *korrigiert* gewährleistet.

Der Vorgang des Herunterladens von Korrekturen wird auf der Seite *Datenaustausch* gestartet. Von dort werden auch fertige Korrekturen auf den WebAssign- Server hochgeladen.