

Fischotter, Melanie; Goedicke, Michael; Kurz-Karaoglu, Filiz; Schwinning, Nils;
Striewe, Michael

Research Report

Erster Jahresbericht zum Projekt "Bildungsgerechtigkeit im Fokus" (Teilprojekt 1.2 - "Blended Learning") an der Fakultät für Wirtschaftswissenschaften

ICB-Research Report, No. 54

Provided in Cooperation with:

University Duisburg-Essen, Institute for Computer Science and Business Information Systems (ICB)

Suggested Citation: Fischotter, Melanie; Goedicke, Michael; Kurz-Karaoglu, Filiz; Schwinning, Nils; Striewe, Michael (2013) : Erster Jahresbericht zum Projekt "Bildungsgerechtigkeit im Fokus" (Teilprojekt 1.2 - "Blended Learning") an der Fakultät für Wirtschaftswissenschaften, ICB-Research Report, No. 54, Universität Duisburg-Essen, Institut für Informatik und Wirtschaftsinformatik (ICB), Essen

This Version is available at:

<https://hdl.handle.net/10419/70906>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



ICB

Institut für Informatik und
Wirtschaftsinformatik

Melanie Fischotter, Michael Goedicke,
Filiz Kurt-Karaoglu, Nils Schwinning, Michael Striewe



Erster Jahresbericht zum Projekt „Bildungsgerechtigkeit im
Fokus“ (Teilprojekt 1.2 – „Blended Learning“) an der Fakul-
tät für Wirtschaftswissenschaften

ICB-RESEARCH REPORT

Die Forschungsberichte des Instituts für Informatik und Wirtschaftsinformatik stellen vorläufige Ergebnisse dar, die i. d. R. noch für spätere Veröffentlichungen überarbeitet werden. Daher sind die Autoren für kritische Hinweise dankbar.

The ICB Research Reports comprise preliminary results which will usually be revised for subsequent publications. Critical comments would be appreciated by the authors.

Die durch das Urheberrecht begründeten Rechte, insbesondere der Übersetzung, des Nachdruckes, des Vortrags, der Vervielfältigung, der Weitergabe, der Veränderung und der Entnahme von Abbildungen und Tabellen – auch bei auszugsweiser Verwertung – bleiben vorbehalten.

All rights reserved. No part of this report may be reproduced by any means, or translated.

Authors' Address:

Melanie Fischotter
Michael Goedicke
Filiz Kurt-Karaoglu
Nils Schwinning
Michael Striewe

Institut für Informatik und
Wirtschaftsinformatik (ICB)
Universität Duisburg-Essen
Universitätsstr. 9
45141 Essen

melanie.fischotter@s3.uni-due.de
goedicke@s3.uni-due.de
filiz.kurt-karaoglu@s3.uni-due.de
nils.schwinning@s3.uni-due.de
michael.striewe@s3.uni-due.de

ICB Research Reports**Edited by:**

Prof. Dr. Heimo H. Adelsberger
Prof. Dr. Frederik Ahleemann
Prof. Dr. Peter Chamoni
Prof. Dr. Klaus Echtele
Prof. Dr. Stefan Eicker
Prof. Dr. Ulrich Frank
Prof. Dr. Michael Goedicke
Prof. Dr. Volker Gruhn
Prof. Dr. Tobias Kollmann
PD Dr. Christina Klüver
Prof. Dr. Klaus Pohl
Prof. Dr. Erwin P. Rathgeb
Prof. Dr. Rainer Unland
Prof. Dr. Stephan Zelewski

Contact:

Institut für Informatik und
Wirtschaftsinformatik (ICB)
Universität Duisburg-Essen
Universitätsstr. 9
45141 Essen

Tel.: +49 201-183-4041
Fax: +49 201-183-4011
Email: icb@uni-duisburg-essen.de

ISSN 1860-2770 (Print)

ISSN 1866-5101 (Online)

Zusammenfassung

Der vorliegende Technical Report beschreibt und diskutiert ein System für Computer Aided Assessments im Bereich Mathematik, welches im Rahmen des Projekts „Bildungsgerechtigkeit im Fokus, Teilprojekt 1.2 - Blended Learning“ realisiert worden ist. Der Report enthält eine Beschreibung des Projekts, eine detaillierte Beschreibung des Systems, erste Erfahrungen und Evaluationen, einen Ausblick auf den weiteren Projektverlauf und einen kurzen Überblick über verwandte Systeme.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Grundlagen und Vorteile zum Blended-Learning	2
1.2	Ausgangslage in der Fakultät für Wirtschaftswissenschaften	3
1.3	Aufbau des Technical Reports	4
2	Projektbeschreibung	5
2.1	Geplante Maßnahmen im gesamten Projekt	5
2.1.1	Inhaltliche Arbeiten (Maßnahmenpaket A)	5
2.1.2	Technologische Arbeiten (Maßnahmenpaket B)	6
2.1.3	Organisatorische Arbeiten (Maßnahmenpaket C)	7
2.2	Ziele des ersten Projektjahres	8
3	Konzept- und Systembeschreibung	10
3.1	Beschreibung von JACK	10
3.1.1	Benutzung durch Studierende	11
3.1.2	Benutzung durch Lehrende	14
3.1.3	Benutzung durch Administratoren	20
3.2	Einsatzszenarien	21
3.3	Integration in umfassende E-Learning-Strukturen	23
4	Architektur und Implementierung	24
4.1	Checker	27
4.1.1	Multiple-Choice-Checker	27
4.1.2	Fill-In-Checker	27
4.2	Variable Aufgabenstellungen	28
4.3	Bilder	29
4.4	L ^A T _E X	30
4.5	Moodle-Integration	30
5	Überblick über Aufgaben und Aufgabentypen	33
5.1	Multiple-Choice-Aufgaben	33
5.1.1	Beispiel: Zweite Binomische Formel	34
5.1.2	Beispiel: Prozentrechnung und Brüche	37

5.2	Fill-In-Aufgaben	38
5.2.1	Beispiel: Kurvendiskussion einer Parabel	38
5.2.2	Beispiel: Lineares Gleichungssystem lösen	40
5.3	Mehrstufige Aufgaben	42
6	Erfahrungen aus dem Vorkurs	45
6.1	Aufbau und Ziele des Vorkurses	45
6.2	Aufgabenerstellung und Qualitätssicherung	46
6.3	Evaluation	48
6.3.1	Evaluationsbögen	48
6.3.2	Nutzerverhalten	52
7	Fazit und Ausblick	57
7.1	Bilanz des ersten Projektjahres	57
7.2	Anstehende Arbeiten im zweiten Projektjahr	57
7.2.1	Kommende inhaltliche Arbeiten	57
7.2.2	Kommende technische Arbeiten	58
7.2.3	Kommende organisatorische Arbeiten	58
A	Überblick über existierende E-Learning-Systeme	59

Abbildungsverzeichnis

3.1	Übersichtsseite von JACK für Studierende	12
3.2	Übersicht eines Kurses für Studierende	13
3.3	Übersichtsseite von JACK für Lehrende und Administratoren	15
3.4	Basiseinstellungen einer Aufgabe	16
3.5	Details einer Aufgabe	17
3.6	Details eines Kurses	19
3.7	Lösungsstatistik für eine Aufgabe	20
3.8	Detailansicht für eine Lösung	21
4.1	Komponentendiagramm für JACK	25
4.2	Ablauf der Prüfung mit Worker	26
4.3	Ablauf der Prüfung in der Geschäftslogik	26
4.4	Ablauf des Aufrufs von JACK durch Moodle 2	32
5.1	Ansicht einer Multiple-Choice-Aufgabe für Studierende	34
5.2	Ansicht einer Multiple-Choice-Aufgabe für Studierende mit Variablen	37
5.3	Ansicht einer Fill-In-Aufgabe für Studierende	39
5.4	Ansicht einer Fill-In-Aufgabe für Studierende mit Variablen	40
5.5	Ansicht Feedback bei einer Fill-In-Aufgabe	42
5.6	Ansicht einer mehrstufigen Aufgabe	43
6.1	Beurteilung des Niveaus des Vorkurses	49
6.2	Beurteilung des Themenspektrums des Vorkurses	49
6.3	Beurteilung des Vorkurses	50
6.4	Benutzung von JACK	50
6.5	Warum JACK nicht häufiger benutzt wurde	51
6.6	Meinungen von Nutzern von JACK	51
6.7	Gesamtpunktzahl der Studierenden im ersten Themengebiet	53
6.8	Mittelwert der eingereichten Lösungen	53

1 Einleitung

Die Universität Duisburg-Essen (UDE) ist eine der größten Universitäten Deutschlands und liegt mitten im Ruhrgebiet. Diese Region hat eine lange Zuwanderungsgeschichte, wodurch Menschen aus verschiedenen Nationen und Kulturkreisen aufeinander treffen. Deshalb verpflichtet sich die UDE nicht nur, Verantwortung für die Forschung, sondern auch für die heterogenen Studierenden zu übernehmen. Über 50% der Studierenden der UDE sind BildungsaufsteigerInnen und fast 25% der Studierenden haben einen Migrationshintergrund. Dies bedeutet für die UDE einen höheren bildungs- und sozialpolitischen Auftrag im Vergleich zu anderen Universitäten.¹

Das Projekt „Bildungsgerechtigkeit im Fokus“ der UDE wird im Rahmen des Bund-Länder-Programms „Qualität der Lehre“ gefördert und hat zum Ziel, die Studienbedingungen und die Qualität der Lehre zu verbessern. Die Studierenden werden vor allem in der Studieneingangsphase mit fakultätsbezogenen Maßnahmen unterstützt und im Verlauf des Studiums werden ihre fachlichen sowie strategischen Kompetenzen durch gezielte und individuelle Betreuung gefördert.²

Das Projekt ist in die drei Maßnahmenbündel „Betreuung und Beratung“, „Potenzialerkennung und -förderung“ und „Qualitätssicherung und strukturelle Instrumente“ mit insgesamt 10 Teilprojekten unterteilt. Eine Beschreibung aller Teilprojekte ist unter <http://www.uni-due.de/bif/aktuelles> verfügbar. Unter dem Maßnahmenbündel „Betreuung und Beratung“ befindet sich u.a. das Teilprojekt 1.2 mit dem Titel „Blended Learning, Selbstlerntools“, an dem die Fakultät für Wirtschaftswissenschaften sowie die Fakultät für Biologie beteiligt sind.

Aus Sicht der Fakultät für Wirtschaftswissenschaften ist die Beteiligung an diesem Teilprojekt insbesondere deshalb attraktiv, da in der Vergangenheit das Fach Mathematik die Studierenden in nahezu allen Studiengängen der Fakultät vor eine besondere Herausforderung gestellt hat. Dies lässt sich durch die unterschiedliche schulische Vorbildung in der Mathematik erklären. Eine deutliche Unterstützung der Studierenden in diesem Bereich erschien daher dringend geboten, um den Studienerfolg insgesamt zu steigern. Da eine massive personelle Aufstockung der Fakultät zur rein manuellen individuellen Betreuung der Studierenden nicht

¹<http://www.uni-due.de/bif/about>

²<http://www.uni-due.de/de/presse/meldung.php?id=2730>

1 Einleitung

realisierbar ist, stellt die Einrichtung eines Blended-Learning-Konzepts gerade in diesem Bereich eine vielversprechende Lösungsmöglichkeit dar.

1.1 Grundlagen und Vorteile zum Blended-Learning

Blended-Learning ist die Verbindung von E-Learning mit der klassischen Präsenzveranstaltung. Durch das E-Learning werden neue und verschiedene Wege der Wissensvermittlung gefunden, welche die Lehrveranstaltungen verbessern sollen. Da der Lernende beim computergestützten Lernen ganz auf sich allein gestellt ist, soll das Blended-Learning diese völlige Isolation kompensieren und der soziale Aspekt des Lernens verstärkt werden. Die Anteile von E-Learning und Präsenzveranstaltungen sind nicht fest definiert. Je nach Bedarf können diese variiert werden.³

Durch den Einsatz von Blended-Learning-Systemen können zahlreiche Vorteile erwartet werden:

- Das System kann zeit- und ortsunabhängig genutzt werden. Einzige Voraussetzungen sind ein Computer und ein funktionierender Internetanschluss. Die Verfügbarkeit von Räumen und Personal, die in der Universität häufig ein limitierender Faktor ist, ist für den erfolgreichen Einsatz mit einer großen Zahl von Studierenden nicht relevant.
- Das System gibt zeitnah nach Einreichung einer Lösung ein individuelles Feedback. Die Studierenden bekommen Fehler sofort aufgezeigt und erhalten die Möglichkeit, die Aufgabe erneut zu bearbeiten.
- Das System kann Studierenden Aufgaben vorschlagen und eine individuelle Führung durch den Lernstoff anbieten. Im Gegensatz zu E-Learning-Systemen, die lediglich Lernmaterial bereitstellen, aber keine Interaktion anbieten, kann das System so zur individuellen Anpassung des Lerntempos beitragen.
- Das System kann Lehrenden Informationen über den Lernfortschritt einer großen Anzahl von Studierenden geben. Dadurch können verbreitete Schwächen aufgedeckt werden, die dann durch zusätzliches Lernmaterial oder zusätzliche Präsenzveranstaltungen angegangen werden können. Ebenso ist es möglich, individuelle Schwächen zu erkennen, auf die dann in einer persönlichen Betreuung eingegangen werden kann.

³Brätz, M.: Einsatz von e-Learning in Vorlesungen und Seminaren, Europäischer Hochschulverlag, 2009, S. 23 ff.

1.2 Ausgangslage in der Fakultät für Wirtschaftswissenschaften

In der Fakultät für Wirtschaftswissenschaften der Universität Duisburg-Essen werden u.a. die Studiengänge Betriebswirtschaftslehre, Volkswirtschaftslehre, Wirtschaftsinformatik, Angewandte Informatik, Lehramt Informatik und Lehramt Wirtschaftswissenschaften gelehrt. In jedem dieser Studiengänge müssen die Studierenden mindestens eine Veranstaltung zur Mathematik erfolgreich absolvieren. Diese Veranstaltungen bestehen meist aus 1-2 Vorlesungen pro Woche sowie einer vorlesungsbegleitenden Übung. Die Studierenden erhalten jede Woche ein Übungsblatt mit Aufgaben, welche den Vorlesungsstoff vertiefen sollen. Diese Aufgaben sind innerhalb einer Woche zu bearbeiten und schriftlich einzureichen. Tutoren korrigieren die eingereichten Übungszettel und rechnen in den Übungen Musterlösungen vor.

Besonders in Vorlesungen mit mathematischen Inhalten lassen sich häufig große Schwierigkeiten der Studierenden mit dem Stoff beobachten, welche in hohen Durchfallquoten bei den Klausuren am Ende dieser Veranstaltungen resultieren. Die Gründe für die Schwierigkeiten sind vielfältig:

Zum einen ist das mathematische Vorwissen der Studierenden, bedingt durch die schulische Ausbildung, oft völlig unterschiedlich. Dies führt zu einer großen Heterogenität unter den Studierenden der ersten Semester. Weniger gut vorgebildete Studierende weisen große Wissenslücken auf und können der Vorlesung oft nicht folgen. Sie erhalten keine Möglichkeit, das fehlende Wissen aufzuholen, denn für zusätzliche Tutorien/Veranstaltungen fehlt es der Universität an Ressourcen.

Zum anderen führen große Mengen von Studienanfängern dazu, dass jeder Tutor viele Studierende gleichzeitig betreuen muss. In den Übungsgruppen kann nur in sehr geringem Maße auf die Bedürfnisse einzelner Studierender eingegangen werden und auch das Feedback durch die korrigierten Übungszettel fällt nicht sehr individuell aus.

Durch die große Anzahl der Lösungen vergeht außerdem zwischen Abgabe und Erhalt der Korrekturen meist eine Woche. Die Studierenden sind in dieser Zeit bereits wieder gezwungen, sich mit einem neuen Übungsblatt - und somit neuen Aufgaben - zu beschäftigen. Dadurch sind ihnen ihre eigenen Lösungen nicht mehr in vollem Maße präsent und sie haben Schwierigkeiten, die Korrekturen nachzuvollziehen.

Den beschriebenen Problemen soll im Rahmen des Projekts mit einem Blended-Learning-System gemäß der oben gegebenen Definition effektiv entgegengewirkt werden.

1.3 Aufbau des Technical Reports

Der vorliegende Technical Report gibt einen Überblick über die technischen und inhaltlichen Arbeiten, die in diesem Teilprojekt an der Fakultät für Wirtschaftswissenschaften im ersten Projektjahr geleistet wurden. Er ist wie folgt gegliedert: Kapitel 2 gibt einen allgemeinen Überblick über den Aufbau des Projekts und die Erreichung der Ziele des ersten Projektjahres. Kapitel 3 erläutert das entwickelte System und seine Ziele auf der konzeptionellen Ebene. Dabei wird insbesondere auch auf verschiedene Einsatzszenarien für das System sowie seine Integration in bestehende E-Learning-Strukturen eingegangen. Kapitel 4 erläutert im Anschluss daran die Umsetzung des Systems aus einem technischen Blickwinkel. Dabei wird ein Überblick über die Systemarchitektur gegeben und einzelne Features des Systems werden herausgegriffen und gesondert betrachtet. Die fachlich-didaktische Auseinandersetzung mit den bisher erarbeiteten Inhalten für das System erfolgt in Kapitel 5, das sowohl auf die bisher realisierten Aufgabentypen als auch auf konkrete Aufgaben eingeht. Bisher gemachte Erfahrungen mit dem Einsatz einer frühen Systemversion werden in Kapitel 6 diskutiert. Dort wird auch eine erste, vorläufige Evaluation der bisherigen Ergebnisse vorgenommen. Der Technical Report schließt mit einem Fazit in Kapitel 7 und erläutert dort auch neue, sich aus den bisherigen Erfahrungen ergebende Projektziele und Arbeitspakete.

Im Verlauf der bisherigen konzeptionellen Arbeiten wurde ein umfassender Vergleich verschiedener existierender E-Learning-Systeme durchgeführt, dessen Ergebnisse zur besseren Übersicht und weiteren Verwendung im Anhang in Kapitel A zusammengefasst werden.

2 Projektbeschreibung

Insgesamt strebt das Teilprojekt die Errichtung eines Blended-Learning-Konzepts mit der engen Verzahnung von persönlicher Betreuung und selbstbestimmtem Lernen unter Zuhilfenahme eines elektronischen Lernsystems mit automatischen und semiautomatischen Mechanismen zur Prüfung von Lösungen an. Dazu sollen sowohl die notwendigen technischen Voraussetzungen geschaffen, als auch eine umfassende Menge an Inhalten erstellt werden, um das Ergebnis erfolgreich im Lehrbetrieb der Fakultät zu implementieren.

2.1 Geplante Maßnahmen im gesamten Projekt

Zur Strukturierung der Projektarbeit über die gesamte Laufzeit des Projekts wurden verschiedene Vorhaben identifiziert und in drei Maßnahmenpakete unterteilt: Maßnahmenpaket A für inhaltliche Arbeiten mit insgesamt vier Einzelmaßnahmen, Maßnahmenpaket B für technologische Arbeiten mit fünf Einzelmaßnahmen und Maßnahmenpaket C für organisatorische Arbeiten mit zwei Einzelmaßnahmen. Als zusätzliche Maßnahme außerhalb der drei Pakete ist eine umfassende Evaluation geplant.

2.1.1 Inhaltliche Arbeiten (Maßnahmenpaket A)

I. Identifikation von Aufgabentypen:

Es werden Aufgabentypen aus verschiedenen Fächern der Studieneingangsphase identifiziert, die im Rahmen des Blended-Learning-Konzepts mit automatischer oder semi-automatischer Prüfung von Lösungen sinnvoll eingesetzt werden können.

II. Identifikation von Kompetenzbereichen:

Es werden Bereiche identifiziert, in denen die Studierenden in der Studieneingangsphase vornehmlich Kompetenzen erwerben sollen bzw. in denen nach derzeitigem Stand besondere Schwierigkeiten mit dem Erwerb grundlegender Kompetenzen bestehen. Identifizierte Aufgabentypen (s.o.) und existierende oder neu zu erstellende Lernmaterialien werden diesen Bereichen zugeordnet.

2 Projektbeschreibung

III. Entwurf und Pflege von Aufgaben:

Zu den identifizierten Aufgabentypen werden konkrete Aufgabensammlungen entworfen, die im elektronischen Lernsystem bereitgestellt und von den Studierenden genutzt werden können. Zu jeder Aufgabe werden auch alle nötigen Bestandteile (z.B. Musterlösungen, Testfälle, Prüfregele, usw.) erstellt, die für die automatische oder semiautomatische Prüfung von Lösungen gemäß der in Maßnahmenpaket B entwickelten Prüfkomponenten notwendig sind. Die Aufgaben werden den identifizierten Kompetenzbereichen zugeordnet und mit initialen Annahmen über die Schwierigkeit versehen, um sie im Rahmen des in Maßnahmenpaket B entwickelten Frameworks für geführtes Lernen einsetzen zu können. Im laufenden Betrieb erfolgt eine ständige Pflege des Aufgabenkatalogs, die insbesondere die Erkenntnisse zum Schwierigkeitsgrad der Aufgaben berücksichtigt, die durch die in Maßnahmenpaket B entwickelte Reportingkomponente ermöglicht werden.

IV. Entwurf eines Tracking-Konzepts mit Steuerungsmaßnahmen:

Es wird ein Konzept entwickelt, wie durch eine Kombination von persönlicher Betreuung der Studierenden und automatisierter Analyse von Ergebnis- und Aktivitätsaufzeichnungen aus dem elektronischen Lernsystem fundierte Aussagen über individuelle Lernfortschritte gewonnen werden können und wie diese Aussagen unmittelbar in den Betreuungsprozess einfließen können. Es wird zudem ein Katalog von Maßnahmen entwickelt, durch die im Rahmen der persönlichen Betreuung individuellen Schwächen der Studierenden effektiv entgegen getreten werden kann, um einen besseren Lernfortschritt zu erzielen. Insbesondere ist mit jeder Maßnahme ein Kriterium verbunden, mit dem die Effektivität der Maßnahme im Nachhinein evaluiert werden kann.

2.1.2 Technologische Arbeiten (Maßnahmenpaket B)

I. Analyse existierender Werkzeuge:

Es werden existierende E-Learning- und E-Assessment-Systeme dahingehend untersucht, wie sie die Zielsetzung des Gesamtprojekts unterstützen können und wie sie ggf. in die zu entwickelnde Lösung integriert werden können.

II. Entwicklung von Prüfkomponenten:

Es werden existierende Systeme geeignet integriert und gegebenenfalls neue Komponenten für das elektronische Lernsystem entwickelt, mit denen Lösungen zu den in Maßnahmenpaket A identifizierten Aufgabentypen automatisch oder semiautomatisch geprüft werden können. Es wird insbesondere angestrebt, Rückmeldungen zu erzeugen, die über eine reine Punktebewertung hinausgehen und somit direkte Hinweise zum Verbessern fehlerhafter Lösungen und zur Selbststeuerung der Studierenden im Lernprozess geben.

III. **Entwicklung eines Frameworks für geführtes Lernen:**

Es wird ein mathematisch-technisches Modell entwickelt und im elektronischen Lernsystem implementiert, das auf Basis der Ergebnisse der Prüfkomponten (s.o.) den Leistungsstand der Studierenden in den in Maßnahmenpaket A identifizierten Kompetenzbereichen sinnvoll messen kann und den Studierenden somit eine Einordnung ihres eigenen Standes im Vergleich zu den Lernzielen anbietet. Insbesondere soll das Modell in der Lage sein, die in Maßnahmenpaket A durchgeführte Zuordnung von Aufgabentypen, Lernmaterialien und Kompetenzbereichen auszuwerten und Studierenden auf dieser Basis individuell sinnvolle nächste Schritte im Lernprozess vorzuschlagen. Das Framework soll sowohl Vorgaben von Lehrenden zur Reihenfolge von Aufgaben und Materialien berücksichtigen können, als auch sinnvolle Reihungen auf Basis bisheriger Ergebnisse dynamisch erzeugen können. Dies schließt insbesondere die kontinuierliche, automatische Neubewertung des Schwierigkeitsgrades von Aufgaben mit ein.

IV. **Entwicklung von Reportingkomponenten:**

Es werden Komponenten für das elektronische Lernsystem entwickelt, mit denen alle notwendigen Daten für das in Maßnahmenpaket A entwickelte Tracking-Konzept erhoben werden können. Außerdem werden Komponenten entwickelt, mit denen die Ergebnisse der automatischen Neubewertung des Schwierigkeitsgrades von Aufgaben (s.o.) analysiert werden können.

V. **Fortlaufende Wartung und Optimierung:**

Die entwickelten Komponenten werden fortlaufend gepflegt und bei Bedarf auf Basis von Einsatzerfahrungen optimiert und erweitert.

2.1.3 Organisatorische Arbeiten (Maßnahmenpaket C)

I. **Schulung der Tutoren:**

Die für die Durchführung der persönlichen, peer-gestützten Betreuung vorgesehenen Tutoren werden entsprechend des Bedarfs umfassend geschult. Es erfolgt sowohl eine didaktische Schulung in Zusammenarbeit mit dem Zentrum für Hochschul- und Qualitätsentwicklung und unter Berücksichtigung der in Maßnahmenpaket A entworfenen Steuerungsmaßnahmen, als auch eine technische Schulung im Umgang mit dem eingesetzten elektronischen Lernsystem.

II. **Errichtung des Tutorennetzwerks:**

Es werden organisatorische Maßnahmen getroffen, um die Anbindung aller Studierenden an die verfügbaren Tutoring-Angebote sicherzustellen und die angestrebten Betreuungsverhältnisse zu erreichen. Ferner werden u.a. durch Foren die Kommunikation von Tutoren und Studierenden auch außerhalb der Tutorien ermöglicht.

2.2 Ziele des ersten Projektjahres

Da die vorgesehene Dauer einiger Maßnahmen mehrere Jahre beträgt und einige Maßnahmen zudem von den Ergebnissen anderer Maßnahmen abhängen, wurden für das erste Projektjahr Zwischenziele festgesetzt. Diese konnten erfolgreich erreicht werden.

- **Aufgabentypen und Kompetenzbereiche sind identifiziert:**

Da der Schwerpunkt des Projekts in der ersten Phase auf den mathematischen Schwächen in der Studieneingangsphase liegen sollte, wurde eine Analyse der Unterrichtsstoffs im Fach Mathematik in der gymnasialen Oberstufe sowie in den Mathematikvorlesungen der Studieneingangsphase vorgenommen. Auf Basis dieser Analyse wurden Kompetenzbereiche benannt und die darin vorkommenden typischen Aufgabentypen identifiziert. Es wurde beschlossen, zunächst generische Aufgabentypen wie Multiple-Choice und Fill-In umzusetzen, da mit diesen die volle Breite der Kompetenzbereiche abgedeckt werden kann. In einem Workshop mit mehreren Lehrstühlen der Fakultät wurde die Liste der Aufgabentypen und Kompetenzbereiche diskutiert und durch weitere Anforderungen und Ideen seitens der zukünftigen potenziellen NutzerInnen des Systems ergänzt.

- **Erste Aufgaben sind entworfen:**

Es wurde ein Fragenpool von ca. 200 Aufgaben erarbeitet, der die o.g. Aufgabentypen nutzt und die meisten Kompetenzbereiche abdeckt. Durch den Einsatz des Fragenpools in einem Vorkurs zum Wintersemester mit 150 Teilnehmern konnten erste Erfahrungen mit den Aufgaben gewonnen werden, die bereits zur Qualitätsverbesserung der Aufgaben ausgewertet wurden.

- **Erste Prüfkomponenten sind in Betrieb:**

Für die oben genannten Aufgabentypen wurden Prüfkomponenten implementiert, so dass der Übungsbetrieb im Vorkurs bereits teilautomatisiert ablaufen konnte. Auch hier konnten bereits erste Erfahrungen zur Qualitätsverbesserung ausgewertet werden.

- **Die Analyse existierender Werkzeuge ist abgeschlossen:**

Die Analyse existierender Werkzeuge wurde in der ersten Jahreshälfte 2012 durchgeführt. Insgesamt wurden 23 Werkzeuge betrachtet - von allgemeinen E-Learning-Plattformen wie z.B. Moodle 2 bis hin zu speziellen Werkzeugen für den Mathematikunterricht wie z.B. Active Math. Diese wurden dahingehend untersucht, ob sie die Anforderungen für ein im Projekt verwendbares Blended-Learning-Werkzeug erfüllen, wie z.B. flexible Feedbackmöglichkeiten. Die Ergebnisse der Analyse sind im Anhang [A](#) dieses Technical Reports zusammengefasst.

- **Der Einsatz einer frühen Systemversion kann beginnen:**

Im o.g. Vorkurs wurde der Übungsbetrieb durch eine frühe Systemversion unterstützt.

Es ist vorgesehen, diese Systemversion und den überarbeiteten Aufgabenpool in Kürze öffentlich zugänglich zu machen. Einzelne beispielhafte Aufgaben verschiedener Typen sind auf dem allgemeinen Demosystem <http://jack-demo.s3.uni-due.de/> verfügbar.

- **Die Schulung von Tutoren kann beginnen:**

Im o.g. Vorkurs waren drei studentische Hilfskräfte als Tutoren im Einsatz, die folglich bereits erste Erfahrungen im Umgang mit dem System, im Stellen von Aufgaben und in der Betreuung von Studierenden sammeln konnten. Diese Erfahrungen können nun ausgewertet werden, um für kommende Einsätze gezielte Schulungen von Tutoren vorzunehmen.

3 Konzept- und Systembeschreibung

Zentraler Inhalt des Teilprojekts ist der Entwurf und die Implementierung eines Selbstlernwerkzeugs, welches den automatisierbaren Teil eines Blended-Learning-Konzepts abdecken kann. Dies bedeutet, dass ein Softwaresystem entstehen soll, welches den Studierenden Aufgaben bereitstellt, Lösungen zu diesen Aufgaben entgegen nimmt, diese Lösungen bewertet und kommentiert und so insbesondere durch hilfreiches und unterstützendes Feedback den Lernfortschritt der Studierenden möglichst individuell fördert.

Wie in Kapitel 2 bereits erwähnt, sollte die Entwicklung eines solchen Systems nach Möglichkeit auf ein bestehendes System aufsetzen. Als Grundlage diene daher das System JACK, das zuvor schon an der Fakultät durch die Arbeitsgruppe „Spezifikation von Softwaresystemen“ entwickelt und betrieben wurde. Dieses kam bisher nur für Programmieraufgaben der Informatik zum Einsatz, ist in seiner Systemarchitektur jedoch bereits auf eine modulare Erweiterbarkeit ausgelegt und damit als Arbeitsgrundlage geeignet.

3.1 Beschreibung von JACK

Das automatisierte Lehr- und Bewertungssystem JACK verfolgt die Zielsetzung, dass Studierende erworbenes Wissen automatisch überprüfen lassen können und zeitnah Rückmeldung vom System erhalten. Zu diesem Zwecke stellt das System den Studierenden Lerneinheiten zur Verfügung, welche die Kenntnisse der Studierenden abfragen. Die kleinsten Lerneinheiten in JACK sind **Aufgaben**. Zu jeder Aufgabe können Lösungen eingereicht werden. Das System überprüft dann die eingereichten Lösungen und gibt den Studierenden Feedback, indem es auf Fehler möglichst individuell eingeht und die Lösungen bewertet. Aufgaben können in Kategorien einsortiert und mit Schlagworten versehen werden. Auch können den Aufgaben Schwierigkeitsgrade zugeteilt werden. Eine weitere Lerneinheit in JACK ist der **Kurs**. Bei einem Kurs handelt es sich um eine vorgegebene Menge von Aufgaben, die zu vorgegebenen Schlagworten passen und deren Schwierigkeitsgrad in einem vorgegebenen Intervall liegt.

In JACK gibt es insgesamt drei verschiedene Nutzerrollen, um die oben aufgeführten Ziele zu gewährleisten:

- **Administratoren:** Der Administrator ist dafür zuständig, das System zu pflegen. Er hat innerhalb des Systems alle Rechte und kann alle Lerneinheiten und Lösungen einsehen.

Er verwaltet auch die NutzerInnen des Systems und kann ihnen die Lehrendenrolle zuteilen.

- **Lehrende:** Der Lehrende erstellt und verwaltet Lerneinheiten und kann Lösungen der Studierenden einsehen. Er kann nur Lerneinheiten verwalten, für welche er Lehrendenrechte besitzt. Dies sind zum einen die von ihm selbst erstellten Lerneinheiten und zum anderen die Lerneinheiten, für welche er von einem anderen Lehrenden oder einem Administrator Lehrendenrechte erhalten hat.
- **Studierende:** Die Studierenden bearbeiten die Lerneinheiten im System. Sie reichen zu den Aufgaben Lösungen ein, welche vom System bewertet werden. Sie bekommen zu ihren Einreichungen eine Rückmeldung und können so ihr Wissen überprüfen. In einem separaten Lösungsüberblick können sie sich alle jemals von ihnen eingereichten Lösungen anzeigen lassen.

3.1.1 Benutzung durch Studierende

Für die Studierenden gibt es zwei verschiedene Systemmodi: Den Standardmodus und den Prüfungsmodus. Normalerweise ist stets der Standardmodus in Betrieb, während der Prüfungsmodus vom Administrator aktiviert werden muss und der Lehrende dann die Möglichkeit hat eine Prüfung zu erstellen.

Meldet sich ein/e NutzerIn im Standardmodus in der Rolle des Studierenden im System an, so erhält er/sie auf der Startseite eine Übersicht aller Kategorien, zu denen Lerneinheiten im System vorhanden sind. Jede Kategorie lässt sich einzeln aufklappen, wodurch die zugehörigen Lerneinheiten sichtbar werden (siehe Abbildung 3.1). Die Studierenden können nun einen Kurs oder eine Aufgabe anwählen und kommen so in die jeweilige Detailansicht dieser Lerneinheit.

Wählen die Studierenden eine Aufgabe, so gelangen sie auf eine neue Seite, deren Gestaltung vom Typ der gewählten Aufgabe abhängt (siehe Abbildung 5.1 auf Seite 34 für eine Multiple-Choice-Aufgabe und siehe Abbildung 5.4 auf Seite 40 für eine Fill-In-Aufgabe). Die im ersten Projektjahr des Teilprojekts relevanten Aufgabentypen stellen auf diesen Seiten jeweils die Fragestellung der Aufgabe dar und fordern zu einer Eingabe auf. Reichen die Studierenden eine Lösung ein, so erhalten sie vom System eine sofortige Bewertung in Form einer Punktzahl sowie einer zu dieser Einreichung passenden Nachricht (siehe Abbildung 5.5 auf Seite 42). Haben sie die Aufgabe falsch gelöst, so können die Studierenden sofort einen neuen Versuch starten. Des Weiteren können sie Hinweise anfordern, falls sie Schwierigkeiten mit der Bearbeitung einer Aufgabe haben und von einem Lehrenden Hinweise zu der gewählten Aufgabe definiert wurden.

3 Konzept- und Systembeschreibung



Fakultät für Wirtschaftswissenschaften >>
 Institut für Informatik und Wirtschaftsinformatik (ICB) >>
Spezifikation von Softwaresystemen | Prof. Dr. Michael Goedicke

UNIVERSITÄT
D U I S B U R G
E S S E N

MENÜ [Eingeloggt als **Jack**]

● Abmelden ? Hilfe

Hauptmenü
 > Lösungsüberblick

English Deutsch

Bereitgestellt durch
FALUNO
 The Ruhr Institute for Software Technology



Gefördert durch

 Bundesministerium
 für Bildung
 und Forschung

Jack ↓

Vorhandene Lerneinheiten

(-) 1 Terme mit einer Variable

Aufgabe/Kurs	Schwierigkeitsgrad
<input type="checkbox"/> Gesetze	1
<input type="checkbox"/> Löse die Betragsgleichung	2
<input type="checkbox"/> Löse die Ungleichung	3
<input type="checkbox"/> Löse die Ungleichung 1	2
<input type="checkbox"/> Lösungsmenge Gleichung	1
<input type="checkbox"/> Oberfläche eines Würfels	1
<input type="checkbox"/> Term vereinfachen und Gleichung lösen	2
<input type="checkbox"/> Term vereinfachen und Gleichung lösen 2	1
<input type="checkbox"/> Terme	1
<input type="checkbox"/> Terme aufstellen	1
<input type="checkbox"/> Terme aufstellen und vereinfachen	1
<input type="checkbox"/> Terme vereinfachen	1
<input type="checkbox"/> Terme zusammenfassen	1
<input type="checkbox"/> Termumformung	1
<input type="checkbox"/> Termumformungen mit Fehlern	1
<input type="checkbox"/> Textaufgabe Terme	1
<input type="checkbox"/> Ungleichung aufstellen und lösen	1
<input type="checkbox"/> Waagschale	1
<input type="checkbox"/> Wasserhöhe im Swimmingpool	1
<input type="checkbox"/> Zahlenrätsel	1

(+) 10 Potenzen

(+) 11 Exponentialfunktionen

(+) 12 Trigonometrie

(+) 2 Terme mit mehreren Variablen

Jack ↓

Vorhandene Lerneinheiten

(-) 1 Terme mit einer Variable

(+) 10 Potenzen

(+) 11 Exponentialfunktionen

(+) 12 Trigonometrie

(+) 2 Terme mit mehreren Variablen

Abbildung 3.1: Screenshot der Übersichtsseite für Studierende, die JACK nach dem Login im Standardmodus anzeigt. Die erste Kategorie ist ausgeklappt und zeigt die enthaltenen Aufgaben an. Die darunter aufgeführten Kategorien sind zugeklappt.

Wählen Studierende einen Kurs, so erhalten sie zunächst eine Übersicht über alle sich in dem Kurs befindenden Aufgaben sowie eine Kursbeschreibung. Bereits bearbeitete Aufgaben werden in dieser Liste durch ein schwarzes Kreuz in der Spalte *Status* neben dem Aufgabentitel markiert (siehe Abbildung 3.2). Die Aufgaben des Kurses können die Studierenden jeweils einzeln anwählen und kommen so zu der oben beschriebenen Detailansicht für Aufgaben. Je nach Einstellungen des Kurses haben sie dort allerdings möglicherweise nicht die Möglichkeit, Hinweise anzufordern oder die Aufgabe nach einem Fehlversuch erneut zu bearbeiten.

The screenshot shows the user interface of the JACK system. At the top left, there is a logo of a stylized person and text identifying the faculty and institute. The main header area includes the university name 'UNIVERSITÄT DUISBURG ESSEN' and a navigation menu with options like 'Abmelden' and 'Hilfe'. The central content area is titled 'Kurse "Trigonometrie"' and contains a table with two columns: 'Status' and 'Aufgaben'. The 'Status' column contains 'x' marks for completed tasks. The 'Aufgaben' column lists various trigonometry topics. On the right side, there are additional navigation options and logos for 'PALUNO' and 'Bildungsgerechtigkeit im Fokus'.

Status	Aufgaben
x	Trigonometrische Funktionen am Einheitskreis
x	Additionssätze
x	Winkel Berechnung
	Cosinus
	trigonometrische Funktion
x	Vereinfache trigonometrischen Term
x	Beziehung zwischen den trigonometrischen Funktionen
	Sinus

Abbildung 3.2: Screenshot der Übersichtsseite eines Kurses in der Studierendenansicht. Die Markierungen in der ersten Spalte geben an, welche Aufgaben innerhalb des Kurses schon bearbeitet wurden.

Auch die Bewertung ihrer Eingabe kann ihnen bis zum Abschluss des Kurses verborgen bleiben.

Jede von den Studierenden gemachte Einreichung wird von JACK in einer Datenbank gespeichert, damit diese jederzeit darauf zugreifen können. Über den Link *Lösungsüberblick* auf der Startseite können sich die Studierenden alle von ihnen bearbeiteten Lerneinheiten nach Datum sortiert anzeigen lassen. Durch einen Klick auf eine dort aufgeführte Lerneinheit kommen sie zu den Lösungsdetails dieser Lerneinheit. In den Lösungsdetails eines Kurses erhalten sie eine Übersicht über alle in dem Kurs eingereichten Lösungen. Ein grünes Häkchen bzw. ein rotes Kreuz rechts neben dem Aufgabentitel symbolisiert, ob die Studierenden die Aufgabe korrekt bearbeitet haben. Zudem befindet sich unter den Lösungen eine Tabelle mit einer Lösungsstatistik, in welcher die Studierenden folgende Angaben zu jeder Aufgabe des Kurses bekommen: Anzahl Bearbeitungen, Anzahl korrekte Bearbeitungen, Durchschnittspunkte, Größte erreichte Punktzahl und Gesamtpunktzahl. Die letzte Zeile der Tabelle errechnet diese Angaben zudem für den gesamten Kurs. Die Lösungsdetails einer Aufgabe entsprechen der Detailansicht der Aufgabe. Die Studierenden sehen die Aufgabenstellung, ihre Lösung und das vom System erhaltene Feedback.

Ist der Prüfungsmodus und mindestens eine Prüfung aktiviert, so müssen Studierende sich mit einer persönlichen TAN einloggen, die vom Lehrenden während der Erstellung einer Prüfung generiert werden kann. Im Prüfungsmodus können die Studierenden lediglich die Menge der vom Lehrenden für diese Prüfung ausgewählten Lerneinheiten sehen und bearbeiten.

3 Konzept- und Systembeschreibung

Ein Zugriff auf den Lösungsüberblick ist erst möglich, wenn die Prüfung wieder deaktiviert wurde. Sollte es während der Prüfung zu einem Ausfall des Rechners kommen, von dem aus der/die Studierende die Prüfung bearbeitet, so kann er/sie sich mit seiner/ihrer TAN an einem anderen Computer am System anmelden und die Prüfung fortsetzen, solange diese aktiviert bleibt.

3.1.2 Benutzung durch Lehrende

Meldet man sich in der Rolle des Lehrenden im System an, so erhält man auf der Startseite ebenfalls eine Übersicht über alle vorhandenen Lerneinheiten, für welche man Lehrendenrechte besitzt (siehe Abbildung 3.3). Die Lerneinheiten sind wie in der Studierendenansicht nach Kategorien sortiert. Wählt man eine der Lerneinheiten an, so kommt man in die Detailansicht dieser Lerneinheit für Lehrende. Des Weiteren hat man die Möglichkeit, eine Lerneinheit zu duplizieren, zu exportieren oder gegebenenfalls zu löschen, wenn noch keine studentischen Lösungen zu dieser Aufgabe vorhanden sind. Im Hauptmenü auf der rechten Seite befinden sich zudem Links, um Lerneinheiten zu erstellen oder zu importieren.

Die Detailansicht einer Lerneinheit zeigt alle zu der Lerneinheit gehörenden Komponenten an. Sie ist gegliedert in mehrere Teilbereiche: Bei Aufgaben gibt es *Basiseinstellungen*, *Tags*, *Ressourcen*, *Zugriffsrechte*, *Checker* und eine *Lösungsstatistik*; bei Kursen *Basiseinstellungen*, *Kurs-Details*, *Zugriffsrechte*, und ebenfalls eine *Lösungsstatistik*. Bis auf die Lösungsstatistik dienen alle Bereiche der Definition der gewünschten Einstellungen für eine Lerneinheit, während der letzte Teilbereich eine Übersicht über die studentischen Ergebnisse bei dieser Lerneinheit liefert. Details zu den einzelnen Teilbereichen finden sich in den folgenden Unterkapiteln.

Basiseinstellungen

In den Basiseinstellungen (siehe Abbildung 3.4 für die Ansicht einer Aufgabe) können ein Titel und optional eine Kategorie für die Lerneinheit festgelegt werden. Beide sind als Texteingabe frei wählbar und dienen der Repräsentation einer Lerneinheit innerhalb der Benutzeroberfläche. Lerneinheiten derselben Kategorie werden in der Darstellung automatisch zusammengefasst. Ebenfalls können als freie Texteingabe eine interne Beschreibung (nur für Lehrende sichtbar) und eine externe Beschreibung (wird dem Lernenden angezeigt) eingegeben werden. Jede neu erstellte Lerneinheit erhält vom System automatisch eine unveränderliche ID, die in den Basiseinstellungen angezeigt wird, und bekommt den erstellenden Lehrenden als Eigentümer zugewiesen. Grundsätzlich haben nur Eigentümer und Administratoren uneingeschränkten Zugriff auf eine Lerneinheit, sofern nicht später weitere Zugriffsrechte definiert werden. Der Zugriff der Lernenden auf eine Aufgabe wird über zwei Optionen gesteuert: Aufgaben können in Kursen und Prüfungen verwendet werden, wenn sie aktiviert sind. Sind sie



Fakultät für Wirtschaftswissenschaften >>
 Institut für Informatik und Wirtschaftsinformatik (ICB) >>
 Spezifikation von Softwaresystemen | Prof. Dr. Michael Goedicke

UNIVERSITÄT
DUISBURG
ESSEN

MENÜ [Eingeloggt als ...]

- Abmelden ? Hilfe
- Hauptmenü**
- Prüfung erstellen
- Aufgabe erstellen
- Aufgabe importieren
- Kurs erstellen
- Kurs importieren
- Administratormenü**
- Benutzer verwalten
- Prüfungsmodus aktivieren
- Lösungsstatistik
- Job-Verwaltung

English Deutsch

Bereitgestellt durch

 The Ruhr Institute for Software Technology



Gefördert durch

 Bundesministerium
für Bildung
und Forschung

Vorhandene Lerneinheiten 🌐

Systemstatus

System läuft im **normalen Modus**, mit **remote**-Authentifizierung.
 Backend ist **nicht aktiv**. Es existieren **0** wartende Backend-Jobs.

Prüfungen

Aufgabe/Kurs

KEINE KATEGORIE ZUGEWIESEN

Titel	Typ	Aktionen
● Graphen lineares Gleichungssystem [copy 1]	MULTISTAGE	[Duplizieren Exportieren]

1 TERME MIT EINER VARIABLE

Titel	Typ	Aktionen
● Die Tangente einer Parabel	FILL_IN	[Duplizieren Exportieren]
● Gesetze	FILL_IN	[Duplizieren Exportieren]
● Löse die Betragsgleichung	MULTISTAGE	[Duplizieren Exportieren]
● Löse die Ungleichung	FILL_IN	[Duplizieren Exportieren]
● Löse die Ungleichung 1	FILL_IN	[Duplizieren Exportieren]
● Lösungsmenge Gleichung	FILL_IN	[Duplizieren Exportieren]
● Oberfläche eines Würfels	MULTISTAGE	[Duplizieren Exportieren]
● Term vereinfachen und Gleichung lösen	FILL_IN	[Duplizieren Exportieren]
● Term vereinfachen und Gleichung lösen 2	MULTISTAGE	[Duplizieren Exportieren]
● Terme	FILL_IN	[Duplizieren Exportieren]
● Terme aufstellen	FILL_IN	[Duplizieren Exportieren]
● Terme aufstellen und vereinfachen	MULTISTAGE	[Duplizieren Exportieren]
● Terme vereinfachen	FILL_IN	[Duplizieren Exportieren]
● Terme zusammenfassen	FILL_IN	[Duplizieren Exportieren]
● Termumformung	FILL_IN	[Duplizieren Exportieren]
● Termumformungen mit Fehlern	MULTIPLE_CHOICE	[Duplizieren Exportieren]
● Textaufgabe Terme	MULTISTAGE	[Duplizieren Exportieren]
● Ungleichung aufstellen und lösen	MULTISTAGE	[Duplizieren Exportieren]
● Waagschale	MULTISTAGE	[Duplizieren Exportieren]
● Wasserhöhe im Swimmingpool	MULTISTAGE	[Duplizieren Exportieren]
● Zahlenrätsel	MULTISTAGE	[Duplizieren Exportieren]

Abbildung 3.3: Screenshot der Übersichtsseite für Lehrende und Administratoren, die JACK nach dem Login anzeigt. Im Menü rechts sind einige Einträge aufgeführt, die nur für Administratoren sichtbar sind.

zudem auch als öffentlich markiert, können die Studierenden direkt von der Übersichtsseite aus auf sie zugreifen.

Ausschließlich bei Kursen kann ein Kursmodus definiert werden. Dieser gibt an, ob während der Bearbeitung der einzelnen Aufgaben des Kurses Hinweise und Feedback angezeigt werden.

Die Basiseinstellungen einer Aufgabe bieten weitere Optionen. Wichtigster Parameter ist hier der sogenannte Sichttyp, der festlegt, wie die Aufgabe in der Web-Oberfläche präsentiert

3 Konzept- und Systembeschreibung

Fakultät für Wirtschaftswissenschaften >>
Institut für Informatik und Wirtschaftsinformatik (ICB) >>
Spezifikation von Softwaresystemen | Prof. Dr. Michael Goedicke

UNIVERSITÄT
DUISBURG
ESSEN

MENU [Eingeloggt als ...] Hilfe

Abmelden

Hauptmenü

- Suche nach Ergebnis
- Suche nach Matrikelnummer
- Fehlerstatistiken
- Lösungen exportieren
- Übersicht

English Deutsch

Bereitgestellt durch
PALUNO
The Ruhr Institute for Software Technology

Bildungsgerechtigkeit im Fokus

Gefördert durch
Bundesministerium für Bildung und Forschung

Aufgabe bearbeiten

Basiseinstellungen

ID der Aufgabe: 3369
Titel: Term vereinfachen und Gleichung lösen
Eigentümer/-in: ...
Kategorie: 1 Terme mit einer Variable
Sichttyp: FILL_IN
Aufgabe ist aktiviert:
Aufgabe ist öffentlich:
Schwierigkeitsgrad (1-5): 2
Interne Beschreibung:
Externe Beschreibung:
Prozentabzug: 0
Minimales korrektes Ergebnis: 50
Evaluierungsregel: #{c3371}

[Einstellungen speichern](#) [Lösung einreichen](#)

Abbildung 3.4: Screenshot des oberen Teils einer Aufgabenseite in der Lehrendenansicht, in der die Basiseinstellungen vorgenommen werden können.

wird. Derzeit sind verschiedene Sichttypen implementiert, die bei Multiple-Choice- und Fill-In-Aufgaben eine direkte Bearbeitung über den Browser ermöglichen und bei Programmieraufgaben Dateien zum Download anbieten sowie die Abgabe per Dateiapload ermöglichen. Ferner kann für eine Aufgabe ein Schwierigkeitsgrad festgelegt werden. Der Schwierigkeitsgrad ist eine ganze Zahl zwischen 1 und 5, wobei 1 für eine sehr einfache und 5 für eine sehr schwierige Aufgabe steht. Auf diese Schwierigkeitsgrade kann bei der Zusammenstellung von Kursen Bezug genommen werden. Es ist geplant, im weiteren Projektverlauf Schwierigkeitsgrade von Aufgaben womöglich automatisiert bestimmen zu können.

Drei weitere Parameter für Aufgaben befassen sich mit deren Bewertung. Grundsätzlich erhält jede Lösung von JACK eine Punktzahl im Intervall von 0 bis 100. Der Lehrende kann angeben, welche Punktzahl mindestens erreicht werden muss, damit die Aufgabe als korrekt gelöst gilt. Außerdem kann er angeben, wie die Wertungen verschiedener Checker (s.u.) miteinander verrechnet werden, falls der Aufgabe mehrere Checker zugeordnet sind, die jeweils unabhängige Bewertungen im Intervall von 0 bis 100 abgeben. Ferner kann der Lehrende bestimmen, dass mehrfache Einreichungen einer Lösung mit Punktabzügen bestraft werden.

The screenshot displays the 'Lehrendenansicht' (Instructor View) for a task. It is divided into several sections:

- Tags:** A section for managing task tags. It includes a 'Hinzufügen' (Add) button, two lists: 'Alle Tags der Aufgabe' (Task Tags) containing 'Algebra', 'Schulwissen', and 'Terme mit einer Variable'; and 'Vorhandene Tags' (Existing Tags) containing 'Ableitungen', 'Algebra', 'Analysis', 'Binomische Formeln', 'Bruchterme', 'Exponentialfunktionen', 'FillIn', 'Flächensätze am Dreieck', 'Geometrie', and 'Geraden'. There are 'Löschen' (Delete) and 'Übernehmen' (Take) buttons.
- Resources:** A table with columns 'Dateiname', 'Größe', 'Typ', and 'Aktionen'. It lists 'Term2.xml' (767 Bytes, WORKING_SHEET) with 'Download' and 'Löschen' (Delete) actions. A 'Ressource hinzufügen' (Add Resource) button is below.
- Lehrende für diese Aufgabe autorisieren:** A section for authorizing instructors. It has a table with columns 'Anmeldename', 'Lehrende(r)', and 'Aktionen'. Two entries are shown, both with 'ja' (yes) and 'Entfernen' (Remove) actions. Below is a dropdown for 'Lehrende(n) für diese Aufgabe autorisieren' (frank.osterbrink) and a 'Lehrende(n) autorisieren' button.
- Checker:** A configuration box for 'InstantFillInChecker (1)'. It shows 'Variablenname: c3371', 'Checker-Name: InstantFillInChecker (1)', and 'Ergebnis-Label: InstantFillInChecker (1) result'. There are checkboxes for 'Zeige Ergebnis in der Übersicht' (checked), 'Zeige Ergebnisdetails' (checked), and 'Checker ist aktiviert' (checked). Buttons include 'Diesen Checker entfernen', 'Lösche alle Ergebnisse von diesem Checker', and 'Konfiguration speichern'. At the bottom, there is a dropdown for 'Checker zu dieser Aufgabe hinzufügen' (Static Java Checker (GreQL)) and a 'Checker hinzufügen' button.

Abbildung 3.5: Screenshot des mittleren Teils einer Aufgabenseite in der Lehrendenansicht, in der Tags, Ressourcen, Zugriffsrechte und Checker verwaltet werden können.

Tags

Tags sind Schlagworte, die man Aufgaben zuordnen kann. Diese sind für die Zusammenstellung von Kursen hilfreich, denn dort kann man eine beliebige logische Verknüpfung von Tags wählen, zu welcher der Kurs Aufgaben anzeigen soll. Nur Aufgaben, welche dieser Verknüpfung entsprechen, werden für den Kurs berücksichtigt.

Die Zuordnung von Tags geschieht über zwei alphabetische Listen (siehe Abbildung 3.5), von denen die eine die bereits der Aufgabe zugeordneten Tags enthält und die andere alle im System vorhandenen Tags, die bereits einmal einer Aufgabe zugeordnet wurden. Über Buttons können ausgewählte Tags zur Aufgabe hinzugefügt bzw. von ihr entfernt werden. Zudem können neue Tags erstellt und der Aufgabe hinzugefügt werden.

3 Konzept- und Systembeschreibung

Ressourcen

Ressourcen sind Dateien, die einer Aufgabe als Arbeitsmaterial zugewiesen sind. Sie stellen die eigentlichen Inhalte der Aufgabe bereit und können unterschiedliche Sichtbarkeiten und Funktionen haben: Als Arbeitsblätter stellen sie den Studierenden Inhalte zur Verfügung und werden durch die Abgabe einer Lösung inhaltlich ergänzt. Als Aufgabenblätter oder Begleitmaterialien stellen sie zusätzliche Inhalte und Informationen zur Verfügung, die von einer Lösung aber nicht verändert werden. Versteckte Materialien sind für Studierende nicht sichtbar, sondern werden zur Konfiguration der Aufgabe und der Überprüfung der Lösung verwendet.

Welche Ressourcen welchen Typs und ggf. mit welchen vorgegebenen Namen an eine Aufgabe angehängt werden können oder müssen, ist vom Sichttyp der Aufgabe abhängig. Bei Multiple-Choice- und Fill-In-Aufgaben muss beispielsweise genau eine Ressource als Arbeitsblatt hochgeladen werden, die sowohl die Fragestellungen und Antwortoptionen für die Studierenden enthält, als auch die komplette Konfiguration für die automatische Überprüfung. Programmieraufgaben verwenden dagegen Arbeitsblätter für Code-Vorlagen, während für die Beschreibung der Aufgabenstellung typischerweise ein Aufgabenblatt zum Einsatz kommt und die Konfiguration der Überprüfungen in versteckten Ressourcen erfolgt.

Zugriffsrechte

Die Einsicht und Bearbeitung von Lerneinheiten ist grundsätzlich nur dem Eigentümer und Administrator gestattet. Lerneinheiten, denen ein Lehrender nicht explizit zugewiesen wurde, werden in dessen Übersicht nicht angezeigt. Um einem Lehrenden den Zugriff auf eine Aufgabe zu gestatten, kann dieser unter der Überschrift „Lehrende für diese Aufgabe autorisieren“ zur Einsicht und Bearbeitung explizit berechtigt werden. Es dürfen alle NutzerInnen autorisiert werden, die in der Benutzerverwaltung als „Lehrende“ freigeschaltet sind.

Checker

Automatische Tests von Aufgaben werden in JACK durch Checker konfiguriert und durchgeführt. Grundsätzlich können alle verfügbaren Checker einer Aufgabe unabhängig von deren Sichttyp zugewiesen werden. Es findet keine automatische Überprüfung statt, ob ein gewählter Checker für den Sichttyp geeignet ist. Allerdings gibt es für bestimmte Aufgabentypen jeweils passende Checker: Den `InstantMultipleChoiceChecker` für Multiple-Choice-Aufgaben und den `InstantFillInChecker` für Fill-In-Aufgaben (siehe Abbildung 3.5). Bei mehrstufigen Aufgaben muss je nach Aufgabenkonfiguration einer oder beide dieser Checker

Kurs-Details

Maximale Anzahl von Aufgaben: 10

Aufgaben haben mindestens Schwierigkeitsgrad: 0

Aufgaben haben höchstens Schwierigkeitsgrad: 5

Kursdauer: 0

Aufgabentypen:

JAVA MULTIPLE_CHOICE UML FILL_IN MULTISTAGE FORMBASED

Tags: Klausuraufgaben

Ableitungen
Algebra
Analysis
Binomische Formeln
Bruchterme
Exponentialfunktionen
FillIn
Flächensätze am Dreieck
Geometrie
Geraden

Vorhandene Tags:

Einstellungen speichern

HINWEIS

Zu Ihrer Konfiguration passen derzeit 10 Aufgaben. Diese sind die folgenden Aufgaben: Logarithmusfunktion, Quadratische Ungleichung, Bestimme die Funktionsgleichungen, Quadratische Funktion, Die Diagonalen des Parallelogramms, Potenzgleichung lösen, Ein bisschen Trigonometrie, Einschränkende Bedingungen bei Wurzeltermen, Ungleichung mit zwei Variablen lösen, Löse das Lineare Gleichungssystem.

Abbildung 3.6: Screenshot des mittleren Teils einer Kursseite in der Lehrendenansicht, in der die Kursdetails festgelegt werden können.

hinzugefügt werden. Weitere Details zu den Checkern und den Aufgabentypen werden in den Abschnitten 4.1, 5.1 und 5.2 diskutiert.

Kurs-Details

In den Kurs-Details (siehe Abbildung 3.6) kann ein Kurs nach den Wünschen des Lehrenden konfiguriert werden. Dazu kann insbesondere ausgewählt werden, wie viele Aufgaben der Kurs enthalten soll und nach welchen Kriterien diese ausgewählt werden. Mögliche Kriterien sind der Schwierigkeitsgrad, die zugewiesenen Tags und der Sichttyp der jeweiligen Aufgabe. Passen zu den gewählten Kriterien mehr Aufgaben, als der Kurs enthalten soll, werden bei jedem Aufruf des Kurses zufällig Aufgaben gewählt. Passen weniger Aufgaben zu den Kriterien, wird dem Lehrenden eine entsprechende Warnung angezeigt.

Lösungsstatistik

In der Lösungsstatistik (siehe Abbildung 3.7 für die Lösungsstatistik einer Aufgabe) werden die Daten über alle bisher eingegangenen Lösungen einer Aufgabe bzw. eines Kurses statistisch zusammengefasst. Blau hervorgehobene Werte sind Links und können angeklickt werden um eine Liste der entsprechenden Datenbankeinträge zu erhalten, von der aus wiederum Details zu einer Lösung abgerufen werden können. Über die beiden Einträge "Suche nach ..." im Hauptmenü kann gezielt nach verschiedenen Kriterien gesucht werden.

3 Konzept- und Systembeschreibung



Lösungsstatistik	
Studenten mit Lösungen:	32
Anzahl der Lösungen:	49
Lösungen CORRECT:	39
Lösungen INCORRECT:	10
Lösungen INTERNAL ERROR:	0
Lösungen WAITING:	0
Kleinster Ergebniswert:	0
Größter Ergebniswert:	100
Median Ergebniswert:	100
Arithmetisches Mittel:	79,59
Standardabweichung:	40,72
Kleinste # an Lösungen:	1
Größte # an Lösungen:	4
Arithmetisches Mittel:	1,53

Alle inkorrekten Ergebnisse löschen | Alle Ergebnisse löschen | Seite aktualisieren
Ergebnisse als Spread-Sheet-Dokument exportieren

Abbildung 3.7: Screenshot des unteren Teils einer Aufgabenseite in der Lehrendenansicht, in der die Lösungsstatistik angezeigt wird.

Die statistische Übersicht ist in vier Blöcke unterteilt. Der erste Block gibt die Anzahl der Studierenden wieder, die eine Lösung eingereicht haben. Der zweite Block bezieht sich auf die Gesamtzahl der eingereichten Lösungen und deren Bewertungsergebnisse. Der dritte Block vermittelt eine Übersicht über die statistischen Kennzahlen der erreichten Ergebniswerte von überprüften Lösungen und der vierte Block konzentriert sich auf die Anzahl von Lösungen pro Studierenden.

Die Detailansicht einer Lösung (siehe Abbildung 3.8) ist für Lehrende ähnlich der Lösungsansicht für Studierende aufgebaut. Sie enthält den Aufgabentext, die studentische Lösung sowie das Feedback. Zudem enthält sie allgemeine Informationen über den Studierenden, wie Matrikelnummer, IP-Adresse und Einreichungszeitpunkt. Zudem sind die Ressourcen dieser Lösung einsehbar, Überprüfungen können erneut angestoßen werden und die Ergebnisse einzelner Checker können je nach Aufgabentyp unterdrückt oder überschrieben werden.

3.1.3 Benutzung durch Administratoren

Die Systemansicht für Administratoren entspricht der für Lehrende, ergänzt um ein Administratormenü unter dem Hauptmenü auf der rechten Seite. Dort kann der Administrator die Benutzerverwaltung vornehmen, d.h. Nutzern die Lehrendenrolle zuteilen, den Prüfungsmodus aktivieren, eine globale Lösungsstatistik für alle Lerneinheiten einsehen und die auf Überprüfung wartenden Lösungen verwalten. Der Administrator kann zudem sämtliche Lerneinheiten im System einsehen und verwalten und dabei auch die Zugriffsrechte auf Lerneinheiten verändern.

The screenshot shows the JACK system interface. At the top left is the logo of the Faculty of Business Sciences and the Institute for Informatics and Business Informatics (ICB). The header includes the university name 'UNIVERSITÄT DUISBURG ESSEN' and a menu with options like 'Abmelden' and 'Hilfe'. The main content area is titled 'Lösungsdetails' and contains several sections: 'Allgemeine Informationen' with fields for 'Matrikelnummer', 'Host-Adresse', 'Aufgabentitel', and 'Einreichung'; 'Ergebnisübersicht' showing a score of 100 and buttons for 'Seite aktualisieren' and 'Alle Ergebnisse löschen'; 'Lösungsressourcen' with a table of files and a download button; and a 'Frage' section with a math problem and an input field for the answer 'x = 12'. Below the question is the 'InstantFillInChecker (1) result' section, which shows a table with columns for 'Position', 'Nachricht', and 'Sichtbarkeit', with the message 'Richtig!' and a 'verbergen' button. The sidebar on the right contains a 'Hauptmenü' with navigation options, language settings (English/Deutsch), and logos for 'PALUNO' and the 'Bundesministerium für Bildung und Forschung'.

Abbildung 3.8: Screenshot der Detailansicht einer Lösung in der Lehrendenansicht.

3.2 Einsatzszenarien

Entsprechend dem Projektziel, den automatisierbaren Teil eines Blended-Learning-Konzepts zu unterstützen, eignet sich das automatisierte Lehr- und Bewertungssystem JACK besonders für das E-Assessment, also die Durchführung und die automatische Auswertung von Lernfortschrittskontrollen. Damit unterscheidet sich JACK stark von Learning-Management-Systemen, welche Inhalte bereitstellen und Instrumente zur Kommunikation zwischen Lehrenden und Lernenden bieten. JACK eignet sich insbesondere auch für das Self-Assessment, d.h. Lernende überprüfen ihr erlerntes Wissen anhand des Systems selbst und erhalten detailliertes Feedback zu ihren Stärken und Schwächen. Dazu ist kein Kontakt zu Dozenten oder Tutoren nötig und es besteht für die Lernenden nicht die Gefahr, sich vor diesen bloßgestellt zu fühlen. Der Einsatz von JACK auf die oben beschriebene Art und Weise ist unter anderem in den folgenden Szenarien denkbar:

1. **Veranstaltungsbegleitung:** In diesem Szenario dient das System zur Unterstützung einer Vorlesung mit begleitendem Übungsbetrieb. Es ist sowohl denkbar, den Übungsbetrieb komplett über JACK abzuwickeln, als auch das System lediglich zur Ergänzung des Übungsbetriebs zu nutzen. Im ersten Fall können Übungsaufgaben ausschließlich in diesem System zur Verfügung stehen und die Abgabe der Aufgaben im System ge-

3 Konzept- und Systembeschreibung

gebenenfalls verpflichtend sein. Im zweiten Fall können ergänzende Übungsaufgaben parallel zu klassischen Übungsformaten bereitgestellt werden, beispielsweise um in Übungsgruppen auf das langwierige Abfragen von Definitionen und vergleichbarem Faktenwissen zu verzichten. Hierfür können stattdessen insbesondere Multiple-Choice-Aufgaben in JACK angeboten werden. Die Lernenden können so selbständig und unabhängig überprüfen, welche Grundlagen sie gut bzw. weniger gut verstanden haben und sich in persönlich betreuten Übungen auf anwendungsorientierte Aufgaben konzentrieren. Je nach didaktischem Konzept der begleiteten Veranstaltung kann den Studierenden in diesem Szenario ein wahlfreier und mehrfacher Zugriff auf Aufgaben gewährt werden, oder die Aufgaben werden in zeitlicher Folge aktiviert und ggf. auch wieder deaktiviert oder nur eine Abgabe pro Aufgabe zugelassen. Im Rahmen des Projektfokus auf Blended-Learning-Konzepte ist dieses Einsatzszenario das Wichtigste.

2. **Prüfungsvorbereitung:** Auch in diesem Szenario besteht ein Bezug zu einer konkreten Lehrveranstaltung, deren Teilnehmer das System auf Prüfungen vorbereitet, indem es prüfungsähnliche Aufgaben oder Verständnisfragen bereitstellt. Durch die Zusammenfassung von mehreren Aufgaben in einem Kurs und die Abschaltung von Feedback in diesem Kurs können dazu auch Klausuren simuliert werden, ohne tatsächlich den Prüfungsmodus von JACK nutzen zu müssen. Über die Optionen des Kurses kann durch die Lehrenden gesteuert werden, welche Aufgaben Bestandteil der simulierten Klausur sind. Somit kann sichergestellt werden, dass Studierende bei der Bearbeitung eine relevante Auswahl aus dem Stoff vorgelegt bekommen und nicht bei freier Wahl der Aufgaben falsche Schwerpunkte setzen. Wie im ersten Szenario kann auch hier der Zugriff auf Aufgaben oder eine simulierte Klausur einmalig oder mehrfach gestattet werden.
3. **Prüfung:** Durch den Einsatz des Prüfungsmodus kann ein Szenario realisiert werden, in welchem den Prüfungsteilnehmern explizit Aufgaben zur Bearbeitung zugewiesen werden. Die Studierenden erhalten zudem kein Feedback und können die erzielten Ergebnisse erst nach Abschluss der Prüfung einsehen. Werden geeignete räumliche Voraussetzungen geschaffen (z.B. Zugriff nur aus einem beaufsichtigten Rechnerpool), können durch dieses Szenario Prüfungen und Testate realisiert werden, die als summative Prüfung der formalen Lernerfolgskontrolle dienen. Typischerweise wird in diesem Szenario nur der einmalige Zugriff auf eine Prüfung gestattet bzw. nur eine Lösung pro Aufgabe und Studierenden gewertet.
4. **Selbstlernkurs:** In diesem Szenario stellt das System unabhängig von einer Lehrveranstaltung Aufgaben oder Kurse zu einem oder verschiedenen Themengebieten bereit. Idealerweise bauen die Aufgaben aufeinander auf und sind zunächst leicht und werden später schwieriger. Sowohl in den Aufgabenbeschreibungen, als auch im Feedback kann auf externes Lernmaterial verwiesen werden, wenn die Aufgaben nicht selbsterklärend sind und die Studierenden zum Selbststudium angeleitet werden sollen. Ein

solches Szenario ist beispielsweise vor Studienbeginn realisierbar, um den Schulstoff zu wiederholen. Passend zur Idee des Selbststudiums wird in diesem Szenario typischerweise der wahlfreie und mehrfache Zugriff auf alle verfügbaren Aufgaben bzw. Kurse gestattet.

5. **Selbsteinschätzung:** Ebenfalls unabhängig von einer Lehrveranstaltung können über das System Aufgaben bereitgestellt werden, die der Selbsteinschätzung der Teilnehmer bzw. der Erhebung vorhandenen Vorwissens dienen. Dieses Szenario ist dem einer (simulierten) Prüfung sehr ähnlich, jedoch mit anderer Zielsetzung. Beispielsweise kann Studienanfängern damit ein Hinweis gegeben werden, ob es sinnvoll wäre einen Brückenkurs zu besuchen. In diesem Szenario brauchen keine Verweise auf Lernmaterialien erstellt zu werden und typischerweise ist nur der einmalige Zugriff auf Aufgaben bzw. Kurse sinnvoll, um Lerneffekte zu vermeiden, die das Ergebnis der Einschätzung verfälschen.

3.3 Integration in umfassende E-Learning-Strukturen

Für ein spezialisiertes E-Assessment-System wie JACK ist eine komfortable Integration in E-Learning-Strukturen und damit insbesondere die Anbindung an Learning-Management-Systeme sowie die flexible NutzerInnen- und Aufgabenverwaltung sehr wichtig. Genauso wie der Einsatz inhaltlich mit anderen Bestandteilen eines Blended-Learning-Angebotes abgestimmt werden muss, um Brüche zwischen Lernmaterialien und (Selbst-)Prüfungsaufgaben zu vermeiden, muss auch die organisatorische und technische Integration so erfolgen, dass keine Brüche an Systemgrenzen entstehen.

Im Rahmen des Projekts kann dabei zum einen auf die bereits vorhandene Anbindung an den zentralen Authentifizierungsdienst der Universität zurückgegriffen werden, der den Studierenden ein Login in JACK mit ihrer Unikennung ermöglicht und über den die Studierenden im System anhand ihrer Matrikelnummer identifiziert werden können. Das System ist damit für die Studierenden leicht zugänglich und ihre Daten können bei Bedarf von den Lehrenden in die üblichen universitären Prozesse eingebunden werden.

Zum anderen erfolgt im Projektverlauf eine enge Anbindung an Moodle 2 (sowie bei Bedarf an weitere Learning-Management-Systeme), mit der die Systemgrenzen sowohl für Studierende als auch für Lehrende noch weiter verwischen. Im Idealfall können Studierende damit durchgängig nur mit Moodle 2 interagieren, das sie bei Bedarf entsprechend der Vorgaben der Lehrenden zu speziellen Aufgaben in JACK weiterleitet, während die Ergebnisse im Gegenzug automatisch wieder an Moodle 2 zurückfließen und dort zum Beispiel in Lernpfaden genutzt werden können.

4 Architektur und Implementierung

Die grundlegende Softwarearchitektur von JACK zerlegt das System in zwei Hauptbestandteile nach dem Master-Worker-Prinzip. Der Master (im Folgenden auch als „Kern“ oder „Server“ bezeichnet) zerfällt dabei wiederum in eine klassische Drei-Schichten-Architektur mit Web-Frontend, Geschäftslogik und Persistenzschicht, während die Worker (im Folgenden auch als „Backend“ bezeichnet) in einer Zwei-Schichten-Architektur aus einer Basis-Komponente und als unabhängige Services realisierten, sogenannten Checkern bestehen. Die genannten Komponenten sind in Abbildung 4.1 dargestellt. In dieser Architektur stellt das Web-Frontend die Benutzerschnittstelle für Lernende, Lehrende und Administratoren bereit. Die Verwendung des Web-Service für Eclipse ist optional und ermöglicht einen vereinfachten Zugriff auf bestimmte Aufgabentypen im Prüfungsmodus für Lernende. Die Erweiterung dieser Schnittstelle um zusätzliche Dienste, die beispielsweise Lehrende beim Erstellen von Aufgaben unterstützen, ist möglich. Der Web-Service für Worker und die Worker-Komponente selber regeln die Kommunikation zwischen dem Server und den Checkern. In der Abbildung ist nur eine Checker-Komponente dargestellt, es kann davon aber beliebig viele geben. Sie realisieren jeweils ein bestimmtes Prüfverfahren für eingereichte Lösungen. Dazu implementieren alle ein bestimmtes Interface, so dass ein Worker jede von ihnen nutzen kann. Je nach Art des Checkers kann dieser selber wieder andere Komponenten benötigen.

JACK ist insgesamt in der Programmiersprache Java implementiert. Die Realisierung des Masters erfolgt als EJB-Anwendung mit einem JSF-Frontend und der JPA für die Persistenzschnittstelle. Das Deployment erfolgt auf einem JBoss 5.1.0 Application Server und für die Datenhaltung kommt eine PostgreSQL 8 Datenbank zum Einsatz. Die Worker werden als OSGi-Anwendungen auf Basis des Equinox-Frameworks realisiert, wo jeder Checker als deklarativer Service und somit unabhängiges Plugin realisiert werden kann.

Jeder Aufgabentyp und jedes Prüfverfahren kann im Rahmen dieser Architektur unabhängig voneinander implementiert und konfiguriert werden. Web-Frontend und Geschäftslogik des Servers stellen dazu allgemeine Methoden für die Handhabung von Aufgaben und Lösungen sowie die zugehörigen Ressourcen bereit. Die speziellen Eigenheiten der einzelnen Aufgabentypen werden dagegen in zusätzlichen Komponenten des Web-Frontends realisiert, die die zu einer Aufgabe bzw. Lösung gehörigen Ressourcen interpretieren und die entsprechenden Eingabemasken und Ausgaben für die Studierenden und Lehrenden generieren. Für die weiter unten in diesem Technical Report vorgestellten Aufgabentypen ist es beispielsweise

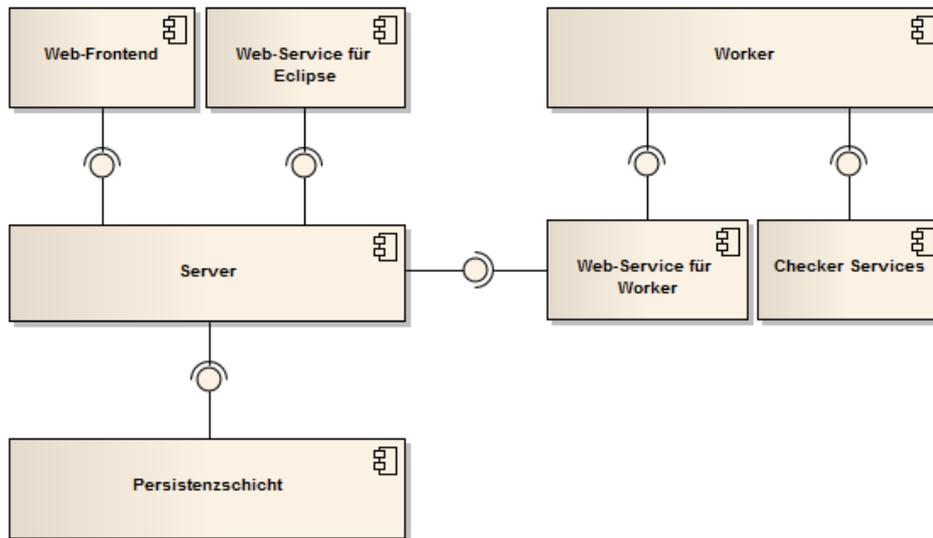


Abbildung 4.1: Komponentendiagramm für JACK. Auf der linken Seite die drei Schichten des JACK-Servers, auf der rechten Seite die zwei Schichten der JACK-Worker. Der Web-Service für Worker ist dem Server zuzurechnen, da er von diesem bereitgestellt wird.

se notwendig, eine XML-Definition der Aufgabenstellung und der zugehörigen Lösungs- und Feedbackoptionen als Ressource an die Aufgabe anzuhängen.

Die Prüfung der Lösungen erfolgt durch die unabhängigen Checker, die wie oben erwähnt als getrennte Komponenten der Worker betrieben oder aber als zusätzliche Komponenten der Geschäftslogik des Servers realisiert werden können. Checker sind dabei grundsätzlich unabhängig vom Aufgabentyp. Ihre Zuordnung erfolgt erst beim Erstellen einer Aufgabe, so dass gleichartigen Aufgaben bei Bedarf verschiedene Prüfverfahren zugewiesen werden können, die durch verschiedene Checker realisiert sind.

Die Entscheidung, ob ein Checker als Komponente der Geschäftslogik oder als Komponente für einen Worker realisiert wird, hängt maßgeblich von der erwarteten Laufzeit einer Überprüfung ab, d.h. vom Rechenaufwand für die Überprüfung der Lösung auf Korrektheit. Checker in der Geschäftslogik werden sofort nach Abgabe der Lösung aufgerufen und sollten möglichst kurze Antwortzeiten haben. Ihre Ergebnisse stehen dann sofort zur Verfügung und können wieder im Web-Frontend angezeigt werden. Checker der Worker sind dagegen für die Durchführung komplexer Prüfverfahren geeignet, bei denen mehr Rechenleistung erforderlich ist. Ihre Ergebnisse stehen daher auch nicht sofort zur Verfügung. Stattdessen wird für jede Lösung und jede so konfigurierte Überprüfung ein Job angelegt. Die Liste der Jobs wird durch den oder die Worker abgearbeitet und das Ergebnis der Lösung hinzugefügt. Erst wenn alle Jobs einer Lösung erledigt sind, kann das Ergebnis durch Studierende oder Lehrende im Web-Frontend eingesehen werden. Durch die parallele Verwendung mehrerer Worker kann eine große Zahl an komplexen Überprüfungen trotzdem mit möglichst kurzen Wartezeiten beantwortet werden. Die unterschiedlichen Abläufe je nach Art der verwendeten Checker sind

4 Architektur und Implementierung

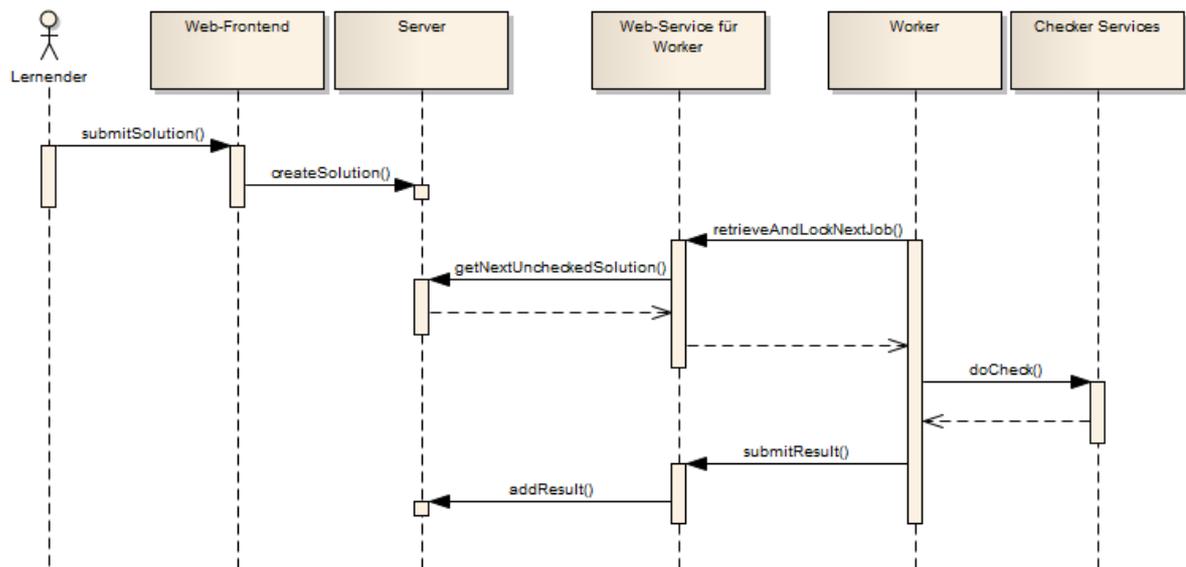


Abbildung 4.2: Ablauf der Einreichung und Prüfung einer Lösung unter Verwendung eines Checker-Services auf einem Worker.

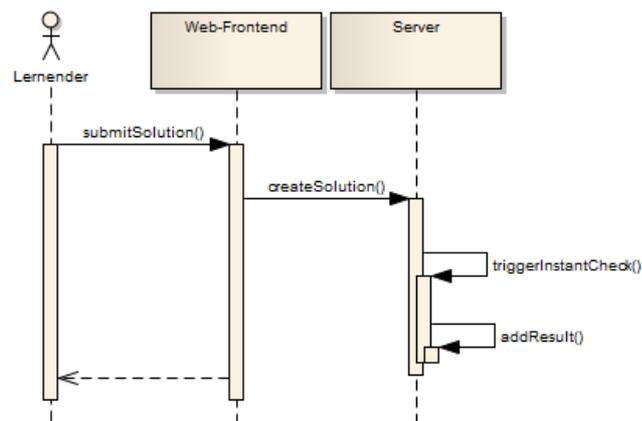


Abbildung 4.3: Ablauf der Einreichung und Prüfung einer Lösung unter Verwendung eines Checkers in der Geschäftslogik.

in den Abbildungen 4.2 und 4.3 als Sequenzdiagramme dargestellt. Bei der Verwendung eines Workers sind zwei Schritte zu erkennen: Zunächst wird die Lösung durch den Lernenden eingereicht und von der Geschäftslogik für die Speicherung vorbereitet. Im zweiten Schritt holt sich der nächste freie Worker die unbearbeitete Lösung ab und reicht die notwendigen Daten an einen passenden Checker-Service weiter, um dessen Ergebnisse wiederum dem Server mitzuteilen. Erst dann kann der Lernende das Ergebnis sehen. Im Gegensatz dazu wird im anderen Fall gleich mit dem Einreichen der Lösung durch den Lernenden die Überprüfung durch einen Checker in der Geschäftslogik angestoßen, so dass das Ergebnis dieses Prozesses dem Lernenden sofort zur Verfügung steht.

4.1 Checker

Im Rahmen des ersten Projektjahres wurden zwei zusätzliche Checker realisiert und in die Geschäftslogik des Servers integriert: Ein Checker zum Prüfen von Multiple-Choice-Aufgaben und ein Checker zum Prüfen von Fill-In-Aufgaben. In beiden Fällen war eine Integration in die Geschäftslogik möglich und sinnvoll, da von sehr kurzen Laufzeiten der nötigen Berechnung auszugehen war und das Ergebnis jeweils als unmittelbare Reaktion des Systems auf eine Eingabe zur Verfügung stehen sollte.

4.1.1 Multiple-Choice-Checker

Die notwendige Berechnung zur Prüfung von Multiple-Choice-Aufgaben beschränkt sich auf die Überprüfung, ob die vom Lernenden angekreuzten Optionen die vom Lehrenden als korrekt markierten Optionen sind. Grundsätzlich kann ein Lehrender beim Erstellen von Multiple-Choice-Aufgaben eine Kombination als richtig auszeichnen und zudem zu beliebig vielen Kombinationen spezifisches Feedback definieren. Aufgrund der tatsächlich angekreuzten Optionen wählt der Checker das passende Feedback aus und vergibt zudem die entsprechende Punktzahl. Weitere Details des Aufgabentyps Multiple-Choice werden in Abschnitt [5.1](#) weiter unten diskutiert.

4.1.2 Fill-In-Checker

Auch im Checker für Fill-In-Aufgaben erfolgt grundsätzlich der Vergleich der Eingabe des Lernenden mit der Vorgabe durch den Lehrenden. Diese kann hier allerdings komplexer ausfallen als im Fall von Multiple-Choice. Der Checker ist daher in der Lage, auch mathematische Ausdrücke auszuwerten und nach Vorgaben des Lehrenden auf Gleichheit oder Ähnlichkeit zu prüfen. Somit ist es beispielsweise möglich, dass nach Vorgabe des Lehrenden sowohl die Eingabe 0.4 , als auch $2/5$ oder $4/10$ sowie weitere entsprechende Brüche als korrekt akzeptiert werden. Die Überprüfung kann allerdings auch so konfiguriert werden, dass nur ein gekürzter Bruch korrekt ist, d.h. $2/5$ als korrekt gewertet wird, nicht aber $4/10$. Zudem kann der Checker auch mit mathematischen Formeln, wie sie z.B. beim Aufstellen einer Funktionsgleichung benötigt werden, umgehen. Er kann erkennen, dass $(x - 2) \cdot (x - 3)$ dem Term $x^2 - 5x + 6$ entspricht und beide Antworten als korrekt werten. Diese Art von Vergleich ist besonders dann sinnvoll, wenn Formeln auf verschiedene Arten eingegeben werden können, denn dann ist es mühselig, alle möglichen korrekten Eingaben zu finden und diese in der Aufgabenstellung auszuzeichnen. Ein Abgleich auf Zeichenketten würde z.B. $x - 2$ und $x - 2$ wegen des Leerzeichens als unterschiedlich werten. Weitere Details des Aufgabentyp Fill-In werden in Abschnitt [5.2](#) weiter unten diskutiert.

Ein wichtiger Aspekt des Checkers für Fill-In-Aufgaben ist der Umgang mit formal inkorrekten Eingaben, beispielsweise aufgrund von Tippfehlern. Damit die Lehrenden für diese Fälle nicht im Einzelnen Feedback definieren müssen, gibt der Checker für bestimmte Eingabefehler generisches Feedback oder nimmt automatisch Fehlerkorrekturen vor. Insbesondere wird bei Dezimalzahlen die Eingabe von Punkt oder Komma gleichwertig behandelt, so dass der Checker diese als gleiche Zahl interpretiert. Die Eingabe $2,5$ ist demnach dasselbe wie die Eingabe 2.5 .

Um die mathematischen Ausdrücke auszuwerten, wurde u.a. die Bibliothek JEval¹ verwendet. Diese kann beispielsweise dynamische und statische Ausdrücke analysieren und bewerten. Weiterhin ermöglicht JEval, die Ausdrucksauswertung von benutzerdefinierten Funktionen, verschachtelten Funktionen und auch variablen Funktionen.

4.2 Variable Aufgabenstellungen

Ein wichtiges Merkmal zahlreicher mathematischer Aufgabentypen ist es, dass dasselbe Aufgabenschema mit verschiedenen Zahlenwerten gefüllt werden kann, um so zum Teil unendlich viele unterschiedliche, aber ähnliche Aufgaben zu generieren. Im Kontext von Blended-Learning-Systemen kann eine solche Funktionalität zudem als motivierender Faktor angenommen werden, da Studierende so einen Anreiz erhalten, bereits bearbeitete Aufgaben noch einmal zu bearbeiten.

Technisch bedeutet diese Anforderung, dass Teile einer Aufgabenstellung als variabel markiert werden können und für diese variablen Stellen vom System Zahlenwerte zufällig gewählt werden. Dabei ist es in der Regel erforderlich, dass sinnvolle Intervalle und Beziehungen zwischen verschiedenen Variablen definiert werden, um zu angemessenen Aufgaben zu kommen. Geschieht dies nicht, können unschöne Aufgaben entstehen, die sich beispielsweise nicht durch ganzzahlige oder rationale Werte lösen lassen, während im schlimmsten Fall sogar sinnlose, bzw. unlösbare Aufgaben entstehen können.

Da die Erzeugung und Verarbeitung von Variablen unabhängig von einem bestimmten Aufgabentyp möglich sein sollte und zudem alle Schritte bei der Verarbeitung einer Aufgabe betrifft (d.h. Stellen der Aufgabe, Prüfen der Lösung, Generierung von Feedback), wurden die entsprechenden Funktionen als Teil der Geschäftslogik von JACK realisiert. Diese liest die Aufgabendefinition aus und ersetzt die variablen Teile durch konkrete Werte, bevor die so verarbeiteten Aufgaben zur Darstellung an das Web-Frontend weitergegeben werden. Analog wird für das von den Checkern erzeugte Feedback verfahren, bei dem vor der Präsentation im Web-Frontend ebenfalls variable Teile durch konkrete Werte ersetzt werden.

¹<http://jeval.sourceforge.net/>

Für jede Variable, die sie in einer Aufgabe definiert haben, können die Lehrenden Vorgaben machen, wie diese ersetzt werden soll: Die erste Möglichkeit besteht darin, einen zufälligen Wert aus einem vorgegebenen Intervall ziehen zu lassen. Die zweite Möglichkeit ist die Vorgabe einer Menge, aus der ein Element zufällig gezogen wird. Diese Möglichkeit ist somit auch für nicht-numerische Variablen geeignet. Die dritte Möglichkeit besteht darin, den konkreten Zahlenwert in Abhängigkeit von anderen, bereits festgelegten Werten, berechnen zu lassen. Auf diese Weise kann sichergestellt werden, dass die Werte einer Aufgabe zueinander passen. Da für die Berechnung nicht nur mathematische Operationen, sondern auch logische Operatoren und Operatoren für Zeichenketten zur Verfügung stehen, kann auch die dritte Option für nicht-numerische Variablen verwendet werden.

Die so bestimmten Werte können nicht nur in der Aufgabenstellung selber, sondern auch in den von den Lehrenden angegebenen Feedback-Optionen der Aufgaben verwendet werden. Dort stehen zusätzlich auch die Eingaben der Studierenden nach demselben Prinzip als Variablen zur Verfügung. Das Feedback zu einer Lösung kann somit in der textuellen Ausgabe sowohl auf die konkreten Werte der Aufgabenstellung als auch die konkreten Eingaben der Studierenden Bezug nehmen.

Ein Beispiel für die Verwendung von Variablen in einer Aufgabe ist in der XML-Definition der Aufgabe in Listing 5.3 auf Seite 41 sowie einem Screenshot dieser Aufgabe in der Studierendenansicht in Abbildung 5.4 auf Seite 40 gegeben.

4.3 Bilder

In der ursprünglichen Version von JACK, die zur Begleitung von Vorlesungen zur Programmierung eingesetzt wurde, handelte es sich bei den Ressourcen, die einer Aufgabe zugewiesen wurden, in der Regel um Programmcodevorlagen und Aufgabenblätter, die von den Studierenden jeweils heruntergeladen werden sollten. Durch die neuen Mathematikaufgaben, die direkt im Browser durch die Studierenden bearbeitet werden sollten, kam Bedarf für zusätzliche Ressourcen hinzu. Insbesondere Bilder, die in der Aufgabe, den Lösungsoptionen oder dem Feedback angezeigt werden sollten, sind für diese Aufgabentypen relevant und können auch außerhalb von mathematischen Aufgaben Anwendung finden.

Nach den allgemeinen Architekturprinzipien von JACK fällt die Behandlung dieser Bilder, die als Ressourcen an die jeweiligen Aufgaben angehängt werden, in den Aufgabenbereich des Web-Frontends, da die korrekte Anzeige der Bilder im Browser ggf. spezifisch für den Aufgabentyp gelöst werden muss. Dementsprechend wurde das Web-Frontend um die nötigen Hilfsmethoden erweitert, welche die passende Einbindung der Bilder sicherstellen. Da Bilder wie alle anderen Ressourcen auch in der Datenbank des Servers gespeichert werden,

```
[img] parallelogramm.png [/img]
```

Listing 4.1: XML-Definition der Bilder

musste zudem ein bereits vorhandenes Servlet so erweitert werden, dass es die Bilder aus der Datenbank auslesen und dem Browser zur Verfügung stellen kann.

Um das Bild anzeigen zu lassen schreibt man in die XML-Datei an der Stelle, wo man das Bild haben möchte, den Ausdruck aus Listing 4.1. Dabei ist allerdings zu beachten, dass man für die Ressource ein Dateiformat wählt, welches vom Browser in Webseiten auch angezeigt werden kann.

4.4 L^AT_EX

Ähnlich wie die Anzeige von Bildern ist auch die Anzeige von mathematischen Formeln eine neue Anforderung an JACK, die sich unmittelbar aus den Projektzielen ergab. Die Umsetzung erfolgt in diesem Fall nicht über eine Änderung am Programmcode, sondern über die Einbindung der JavaScript-Library *mathjax*² in der vom Web-Frontend erzeugten Ausgabe. Diese rendert automatisch alle Inhalte der ausgegebenen Webseiten, die in $\$$ -Zeichen eingeschlossen sind, in L^AT_EX. Dies entspricht dem üblichen Vorgehen, wie L^AT_EX-Code auch in Fließtextdokumenten verwendet wird. Für Lehrende, die im Umgang mit L^AT_EX vertraut sind, sind Aufgaben mit mathematischen Formeln dementsprechend ähnlich einfach zu erstellen wie klassische Übungszettel oder Papierklausuren.

Ein Beispiel für die Verwendung von L^AT_EX in einer Aufgabe ist in der XML-Definition der Aufgabe in Listing 5.1 auf Seite 35 sowie einem Screenshot dieser Aufgabe in der Studierendenansicht in Abbildung 5.1 auf Seite 34 gegeben.

4.5 Moodle-Integration

Eine weitere Erweiterung von JACK, die im Rahmen des Projekts realisiert wurde, ist die Anbindung an *Moodle* 2³. Moodle 2 ist ein weitverbreitetes Open-Source Learning-Management-System, welches auch die UDE einsetzt. Dieses Learning-Management-System erlaubt es dem Lehrenden virtuelle Klassen bzw. Kurse zu erstellen und diese mit vielen Lernmaterialien, wie z.B. Arbeitsblättern, Videos etc. zu versorgen. Weiterhin kann der Lehrende mit den Lernenden kommunizieren, und auch die Lernenden untereinander können miteinander kommunizieren, wie z.B. durch Chats, Privatnachrichten und in Foren. Außerdem können Materialien

²<http://www.mathjax.org/>

³<https://moodle.org/>

Erster Jahresbericht zum Projekt "Bildungsgerechtigkeit im Fokus" (Teilprojekt 1.2 - "Blended Learning") an der Fakultät für Wirtschaftswissenschaften und Aktivitäten in Lernpfaden organisiert werden, in denen ein Schritt erst nach Abschluss eines vorherigen Schrittes aktiviert wird.

Andererseits bietet die Moodle 2 Instanz der Universität UDE keine Unterstützung für \LaTeX , keine Rechenfähigkeiten für mathematische Aufgaben, keine variablen Aufgabenstellungen und enger begrenzte Feedbackmöglichkeiten als JACK. Um die Vorteile beider Systeme trotzdem miteinander kombinieren zu können, wurde eine Anbindung von JACK an Moodle 2 realisiert. Somit besteht die Möglichkeit, dass der Lehrende die Lernenden direkt von Moodle 2 nach JACK weiterleiten kann. Dazu muss durch den Lehrenden eine Aktivität vom Typ „externes Tool“ angelegt werden, in der unter anderem eine ID der gewünschten Lerneinheit in die Konfiguration eingetragen wird. Diese Aktivität wird innerhalb von Moodle 2 als Link angezeigt. Wenn der Lernende sich in Moodle 2 einloggt und auf den Link klickt, wird automatisch die Verbindung zu JACK aufgebaut, der Lernende in JACK eingeloggt und die Lerneinheit im selben Fenster angezeigt. Dort kann dann der Lernende diese dann bearbeiten.

Die Integration folgt dabei dem IMS-LTI-Standard⁴ und ist damit grundsätzlich in der Lage, auch die Anbindung an andere Learning-Management-Systeme zu gewährleisten. Im ersten Projektjahr wurde zunächst nur die Kommunikation von Moodle 2 zu JACK realisiert. Die Rückmeldung von Ergebnissen aus JACK an Moodle 2, die beispielsweise für die volle Nutzung der Lernpfade notwendig ist, soll im zweiten Projektjahr realisiert werden.

Der bisher implementierte Ablauf ist in Abbildung 4.4 beispielhaft für den erfolgreichen Aufruf einer Aufgabe als Sequenzdiagramm dargestellt. In einer ersten Anfrage sendet dabei Moodle 2 als POST-Request den Nutzernamen des in Moodle 2 eingeloggten Nutzers und die ID der gewünschten Aufgabe. JACK validiert die Anfrage mithilfe von *OAuth*⁵ und loggt den/die NutzerIn in JACK ein. Im konkreten Anwendungsszenario der UDE bedeutet dies, dass für den Nutzernamen aus Moodle 2 eine gültige Matrikelnummer vom Authentifizierungsdienst der Universität abgefragt wird. Kann keine Matrikelnummer bezogen werden oder schlug zuvor bereits die Validierung der Anfrage fehl, bricht der Loginvorgang mit einer Fehlermeldung ab und der/die NutzerIn wird nicht zur gewünschten Aufgabe weitergeleitet. Gelingt das Login, speichert JACK die ID der gewünschten Aufgabe ab, lädt die Aufgabe aber noch nicht vollständig, sondern leitet Moodle 2 zunächst zu einer generischen Redirect-Seite weiter. Moodle 2 ruft diese Seite über einen GET-Request auf, woraufhin JACK die angeforderte Aufgabe nun auch tatsächlich lädt. Erst dann kann JACK entscheiden, welche Seite zur Darstellung dieser konkreten Aufgabe angezeigt werden soll, so dass ein zweiter Redirect erfolgt, der automatisch im Browser des Nutzers durchgeführt wird und endgültig zur gewünschten Seite führt.

⁴<http://www.imsglobal.org/lti/index.html>

⁵<http://oauth.net/>

4 Architektur und Implementierung

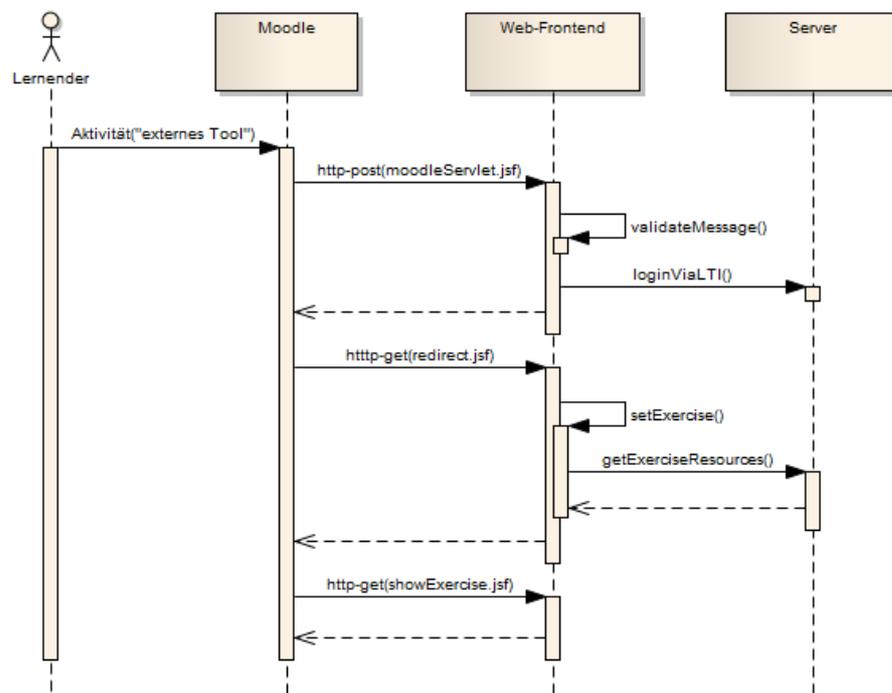


Abbildung 4.4: Ablauf des Aufrufs von JACK durch Moodle 2 entsprechend des IMS-LTI-Standards.

5 Überblick über Aufgaben und Aufgabentypen

Im ersten Projektjahr wurde ausschließlich mit zwei Aufgabentypen gearbeitet: „Multiple-Choice“ und „Fill-In“. Zudem wurden mehrstufige Aufgaben entworfen, bei denen jeder Schritt von einem der vorgenannten Typen ist. Jede Aufgabenstellung samt der zugehörigen Lösungs- und Feedbackmöglichkeiten muss vom Lehrenden in einer XML-Datei konfiguriert werden, die als Ressource an die Aufgabe angehängt wird.

5.1 Multiple-Choice-Aufgaben

Eine Multiple-Choice-Aufgabe besteht aus einer Frage und einer beliebig großen Menge von Antwortoptionen, von denen keine, eine oder mehrere richtig sein können. Das Ziel des Lernenden ist es, die richtige/n Lösung/en herauszufinden und auszuwählen. Da die Möglichkeit besteht, nur eine Antwort als richtig zu markieren, schließt dieser Aufgabentyp auch zusätzlich den Aufgabentyp Single-Choice mit ein. Durch Multiple-Choice-Aufgaben kann das Faktenwissen der Lernenden überprüft werden, jedoch nicht die Kreativität. Im Allgemeinen eignen sich Multiple-Choice-Aufgaben gut dazu, in den Vorlesungen erworbenes Faktenwissen, wie z.B. Definitionen oder Theoreme abzuprüfen. Dazu sollte man die falschen Antwortoptionen, die auch als "Distraktoren" bezeichnet werden, mit viel Sorgfalt auswählen. Sie sollten für einen Lernenden, welcher das zu überprüfende Wissen nicht beherrscht, den Anschein erwecken korrekt zu sein. Dies kann z.B. geschehen, indem man häufig auftretende Fehler in den Distraktoren abbildet. Der Lehrende kann einstellen, dass die Antwortoptionen vor der Anzeige gemischt werden, sodass bei erneutem Auswählen der Aufgabe eine andere Reihenfolge der Antworten entsteht. So lässt sich erreichen, dass der Lernende sich mit den Inhalten der Antworten auseinandersetzen muss und sich nicht bloß die Position der korrekten Antworten zu merken braucht. Durch die Eingabe aller richtigen Antwortmöglichkeiten bekommt der Lernende die volle Punktzahl. Eine Möglichkeit der Vergabe von Teilpunkten gibt es zurzeit nicht, sie ist jedoch bereits in Planung. Der Lehrende kann für jede Kombination von ausgewählten Antwortoptionen ein individuelles Feedback einstellen. So kann er auf jeden Fehlerfall genau eingehen und dem Lernenden beschreiben, wo seine Fehler liegen. Da es jedoch bei n Antwortoptionen 2^n mögliche Lösungen gibt, kann der Lehrende ähnliche Kombinationen bündeln und ein gemeinsames Feedback zu diesen verfassen. Ist die Lösung des Lernenden falsch, so bekommt er das für diesen Fehlerfall vorgesehene Feedback angezeigt und hat die Möglichkeit, die Aufgabe zu wiederholen. Außerdem kann der Lehrende

5 Überblick über Aufgaben und Aufgabentypen

Fakultät für Wirtschaftswissenschaften >>
Institut für Informatik und Wirtschaftsinformatik (ICB) >>
Spezifikation von Softwaresystemen | Prof. Dr. Michael Goedicke

UNIVERSITÄT
DUISBURG
ESSEN

MENÜ [Eingelogg als ...]

Abmelden Hilfe

Hauptmenü
> Übersicht

English Deutsch

Bereitgestellt durch
PALUNO
The Ruhr Institute for Software Technology

Bildungsgerechtigkeit im Fokus

JACK (Version 2.3.1) | © 2009-2012 Spezifikation von Softwaresystemen

Abbildung 5.1: Screenshot einer Multiple-Choice-Aufgabe in der Studierendenansicht. Die Formeln werden in \LaTeX gerendert und die Antwortoptionen können bei wiederholtem Aufruf der Aufgabe in anderer Reihenfolge stehen. Die XML-Definition der Aufgabe ist in Listing 5.1 aufgeführt.

eine beliebige Anzahl von Hinweisen zu jeder Aufgabe einstellen, welche der Lernende über den „Hinweis“-Button einholen kann. Mit jedem Klick auf diesen Button wird ein Hinweis mehr angezeigt, bis alle Hinweise zur Verfügung stehen. Idealerweise wird der Lernende so zu der richtigen Lösung der Aufgabe hingeführt.

Im Folgenden werden einige Beispielaufgaben zum Aufgabentyp Multiple-Choice diskutiert. Dabei wird auch vor allem auf die einzelnen Feedbackfälle eingegangen. Um eine Lösung zu diskutieren, wird ein Muster bestehend aus 0 und 1 verwendet, wie es auch zur Konfiguration der Aufgabe in der XML-Datei angegeben werden muss. Dabei steht eine 1 für eine ausgewählte Antwort und eine 0 für eine nicht ausgewählte Antwort. Das Muster muss immer genau so viele Zeichen haben, wie Antwortoptionen vorgegeben sind. Das Muster 0110 steht also zum Beispiel für die Lösung, bei der die zweite und die dritte Antwortoption ausgewählt wurden. Ist es nicht relevant, ob eine Antwortoption ausgewählt wurde, so wird das Zeichen * dafür verwendet. Das Muster *110 fasst also die beiden Muster 0110 und 1110 zusammen.

5.1.1 Beispiel: Zweite Binomische Formel

Abbildung 5.1 zeigt eine Aufgabe zur zweiten binomischen Formel. Die zugehörige XML-Definition der Aufgabe ist in Listing 5.1 wiedergegeben. Das Muster für die richtige Lösung bei dieser Aufgabe ist offensichtlich 0101. Antwort 1 und 3 bilden häufig gemachte Fehler bei

```

<?xml version="1.0" encoding="iso-8859-1"?>

<exercise>
  <task> Welche der folgenden Terme entsprechen  $(a-b)^2$ ? </task>
  <answers randomize="true">
    <option>  $a^2-b^2$  </option>
    <option>  $(b-a)^2$  </option>
    <option>  $a^2-2ab-b^2$  </option>
    <option>  $b^2-2ab+a^2$  </option>
  </answers>
  <feedback>
    <choice pattern="0000"> Diese Antwort ist leider falsch. Multipliziere den Term aus, indem du Add "Distributivgesetz" to dictionarysetz anwendest:  $(a-b)^2 = (a-b)(a-b) = a(a-b) - b(a-b) = \dots$  </choice>
    <choice pattern="1000"> Diese Antwort ist leider falsch. Multipliziere den Term aus, indem du das Distributivgesetz anwendest:  $(a-b)^2 = (a-b)(a-b) = a(a-b) - b(a-b) = \dots$  </choice>
    <choice pattern="0100"> Diese Antwort ist richtig, es gibt jedoch noch eine weitere richtige Antwort. Wie sieht der Term aus, wenn du ihn ausmultiplizierst? </choice>
    <choice pattern="0010"> Diese Antwort ist leider falsch, denn  $(-b)^2=b^2$ .</choice>
    <choice pattern="0001"> Du hast die binomische Formel richtig angewendet. Es gibt aber eine weitere richtige Antwort. </choice>
    <choice pattern="1100"> Diese Antwort ist leider falsch. Multipliziere den Term aus, indem du das Distributivgesetz anwendest:  $(a-b)^2 = (a-b)(a-b) = a(a-b) - b(a-b) = \dots$  </choice>
    <choice pattern="1010"> Diese Antwort ist leider falsch. Die beiden Antworten widersprechen sich. Multipliziere den Term aus, indem du das Distributivgesetz anwendest:  $(a-b)^2 = (a-b)(a-b) = a(a-b) - b(a-b) = \dots$  Wie lautet das Vorzeichen von  $b^2$ ?</choice>
    <choice pattern="1001"> Diese Antwort ist leider falsch. Multipliziere den Term aus, indem du das Distributivgesetz anwendest:  $(a-b)^2 = (a-b)(a-b) = a(a-b) - b(a-b) = \dots$  </choice>
    <choice pattern="0110"> Diese Antwort ist leider falsch. Achte beim Ausmultiplizieren auf das Vorzeichen. </choice>
    <choice pattern="0011"> Beide Antworten machen zusammen keinen Sinn.</choice>
    <choice pattern="1110"> Mindestens zwei der Antworten widersprechen sich. </choice>
    <choice pattern="1101"> Mindestens zwei der Antworten widersprechen sich. </choice>
    <choice pattern="1011"> Mindestens zwei der Antworten widersprechen sich. </choice>
    <choice pattern="0111"> Mindestens zwei der Antworten widersprechen sich. </choice>
    <choice pattern="1111"> Mindestens zwei der Antworten widersprechen sich. </choice>
    <choice pattern="andere"> Multipliziere  $(a-b)^2$  schrittweise aus. Schreibe  $(a-b)^2 = (a-b) \cdot (a-b) = c \cdot (a-b)$ , wobei selbstverständlich  $c = (a-b)$  gilt. Wende nun das Distributivgesetz an, ersetze  $c$  wieder durch  $(a-b)$  und wende dann das Distributivgesetz erneut an. </choice>
  </feedback>
  <correctanswer pattern="0101"> Richtig. </correctanswer>
  <advice>
    <option>Multipliziere  $(a-b)^2$  schrittweise aus. Schreibe  $(a-b)^2 = (a-b) \cdot (a-b) = c \cdot (a-b)$ , wobei selbstverständlich  $c = (a-b)$  gilt. Wende nun das Distributivgesetz an, ersetze  $c$  wieder durch  $(a-b)$  und wende dann das Distributivgesetz erneut an. Achtung: Es ist mehr als eine Antwort korrekt! </option>
    <option>Überprüfe deine Antworten an einem Beispiel. Wähle z.B.  $a = 2$  und  $b = 1$ .</option>
  </advice>
</exercise>

```

Listing 5.1: XML-Definition der Multiple-Choice-Aufgabe aus Abbildung 5.1.

5 Überblick über Aufgaben und Aufgabentypen

der zweiten binomischen Formel ab, nämlich dass man beim Ausmultiplizieren den gemischten Term $-2ab$ vergisst bzw. die Berechnung $(-b) \cdot (-b)$ nicht korrekt ausführt. Zudem gilt es zu erkennen, dass hier zwei Antwortoptionen richtig sind. Üblicherweise wird in der Schule die Gleichheit $(a-b)^2 = a^2 - 2ab + b^2$ besprochen. Daher erkennen die Lernenden häufig 4. als richtige Antwort. Allerdings erkennen sie oft nicht, dass wegen $(-x)^2 = x^2$ auch 2. korrekt ist. Insgesamt wurden vom Autor der Aufgabe sechs verschiedene Feedbacknachrichten zu der Aufgabe erstellt, welche im Folgenden im Detail diskutiert werden:

1. Wird das Muster 0001 eingereicht, so hat der Lernende eine richtige Antwort erkannt und die Andere übersehen. Er wird daher durch das Feedback "Du hast die binomische Formel richtig angewendet. Es gibt aber noch eine weitere richtige Antwort" darauf hingewiesen.
2. Wird das Muster 0100 eingereicht, so hat der Lernende ebenfalls nur eine von zwei richtigen Antworten erkannt. Er wird durch das Feedback "Diese Antwort ist richtig, es gibt jedoch noch eine weitere richtige Antwort. Wie sieht der Term aus, wenn du ihn ausmultiplizierst?" dazu motiviert, die Klammern aufzulösen.
3. Die Antworten 1, 3 und 4 widersprechen sich jeweils paarweise. Daher wird der Lernende durch das Feedback "Mindestens zwei der von dir ausgewählten Antworten widersprechen sich" auf diese Tatsache hingewiesen, falls er mindestens zwei dieser drei Antwortoptionen zusammen auswählt. Wählt er nur die beiden falschen Antwortoptionen aus, also 1 und 3, so wird ihm zusätzlich noch mitgeteilt, dass nicht $(-b)^2 = -b^2$ gilt, da $-b^2$ in beiden Antworten vorkommt.
4. Wird das Muster 0010 eingereicht, so wird dem Lernenden die Nachricht "Diese Antwort ist leider falsch, denn es gilt $(-b)^2 = b^2$." übermittelt, welche ihm mitteilt, wo sein Fehler liegt.
5. Reicht der Lernende das Muster 0110 ein, so hat er eine richtige Antwort gefunden, aber beim Ausmultiplizieren einen Fehler gemacht. Dies wird ihm durch das Feedback "Achte beim Ausmultiplizieren auf das Vorzeichen" mitgeteilt.
6. In allen anderen Fällen wird dem Lernenden noch mal erläutert, wie er den Term umformen kann: "Multipliziere den Term aus, indem du das Distributivgesetz anwendest: $(a-b)^2 = (a-b)(a-b) = a(a-b) - b(a-b) = \dots$ ".

Die diskutierte Aufgabe war auch Bestandteil des Vorkurs vor dem Wintersemester 12/13 (siehe Kapitel 6). Es wurden insgesamt 48 Lösungen von 21 verschiedenen Lernenden eingereicht. Davon waren 21 Lösungen korrekt und 27 Lösungen falsch. Von den 21 Lernenden reichten 13 zunächst eine falsche Lösung ein. 12 dieser 13 Lernenden versuchten sich daraufhin erneut an der Aufgabe und 10 schafften es auch später, die richtige Lösung zu finden. Der oben diskutierte Fall, dass nur 4. als Antwort gewählt wird, nicht jedoch zusätzlich auch 2.

Fakultät für Wirtschaftswissenschaften >>
Institut für Informatik und Wirtschaftsinformatik (ICB) >>
Spezifikation von Softwaresystemen | Prof. Dr. Michael Goedicke

UNIVERSITÄT
DUISBURG
ESSEN

MENÜ [Eingelogg als ...]

Abmelden Hilfe

Hauptmenü
Übersicht

English Deutsch

Bereitgestellt durch
PALUNO
The Ruhr Institute for Software Technology

Bildungsgerechtigkeit im Fokus

Gefördert durch
Bundesministerium für Bildung und Forschung

JACK (Version 2.3.1) | © 2009-2012 Spezifikation von Softwaresystemen

Abbildung 5.2: Screenshot einer Multiple-Choice-Aufgabe mit variabler Aufgabenstellung in der Studierendenansicht.

war dabei in der Tat der häufigste Fehler. Diese Lösung kam unter den 27 falschen Lösungen 12 Mal vor und war die mit Abstand am häufigsten eingereichte falsche Lösung.

5.1.2 Beispiel: Prozentrechnung und Brüche

Abbildung 5.2 zeigt eine Multiple-Choice-Aufgabe mit Variablen. Das bedeutet, bei erneuter Anwahl der Aufgabe verändern sich die Zahlenwerte in der Aufgabenstellung und auch bei den Antwortoptionen. Der Lernende soll bei dieser Aufgabe die Brüche finden, welche einer vorgegebenen Prozentzahl p entsprechen. Von den vier zur Auswahl stehenden Antworten sind jeweils zwei richtig und zwei falsch. Dabei handelt es sich bei den korrekten Antworten je einmal um den Bruch $\frac{p}{100}$ in gekürzter und einmal in erweiterter Form. Damit sich dieser Bruch überhaupt kürzen lässt, wird $p = 4 \cdot k$ gewählt, wobei k eine zufällig gewählte natürliche Zahl mit $2 \leq k \leq 12$ und $k \neq 5$ ist. Anschließend wird eine Zahl $m \in \{2, 4\}$ gewählt, welche sowohl p als auch 100 teilt, um den Bruch zu kürzen. Zudem wird eine korrekte Antwort bestimmt, indem $\frac{p}{100}$ geeignet erweitert wird. Als Distraktoren fungieren bei dieser Aufgabe zum einen der Bruch $\frac{1}{p}$ und zum anderen der Bruch $\frac{1}{n}$, wobei n die größte natürliche Zahl kleiner als $\frac{100}{p}$ ist. Man beachte, dass dieser Bruch keine richtige Lösung darstellen kann, da die Zahl p so gewählt wurde, dass sie 100 nicht teilt. Auch bei dieser Aufgabe verändert sich die Reihenfolge der Antwortoptionen vor jeder Anzeige. Für diese Aufgabe wurden insgesamt

5 Überblick über Aufgaben und Aufgabentypen

vier Feedbackfälle für falsche Antworten generiert. Man kann hier die Fälle unterscheiden, in denen der Lernende eine falsche Antwort auswählt und diejenigen in denen der Lernende eine richtige Antwort vergisst. Außerdem gibt es noch ein Feedback für den Fall, dass der Lernende keine Antwort als richtig erkennt.

5.2 Fill-In-Aufgaben

Fill-In-Aufgaben werden im Deutschen als Lückentext-Aufgaben bezeichnet. Zusätzlich zu der Aufgabenstellung gibt es ein oder mehrere Fill-In Kästchen, in welche die Lösungen eingetragen werden müssen. Die Anzahl und die Größe der Kästchen kann der Lehrende in der Aufgabenstellung selbst einstellen. Dabei sollte er darauf achten, die Größe von der Länge der zu erwartenden Antwort abhängig zu machen. Außerdem sollte beachtet werden, dass die Größe des Kästchens auch ein Hinweis für die Lernenden sein kann, ob eine kurze oder lange Eingabe erwartet wird.

Mit den Fill-In-Aufgaben können Texte, Zahlen und auch Formeln abgefragt werden. Für jedes Eingabeformat besitzt der zugehörige Checker Funktionen, um die eingegebene Lösung mit der richtigen Lösung zu vergleichen. Das Feedback des Systems kann auch bei diesem Aufgabentyp abhängig von der eingereichten Lösung gemacht werden. Dazu muss für jeden Feedbackfall ein boolescher Ausdruck geschrieben werden mit Platzhaltern für die Eingaben des Lehrenden. Der Fill-In-Checker setzt die Eingaben des Lernenden und die Variablen (falls vorhanden) ein und wertet den Ausdruck anschließend aus. Ist der Ausdruck wahr, so wird das für diesen Fall vorgesehene Feedback zurückgegeben. Zudem ist es möglich, mit der Lösung des Lernenden im Feedback zu rechnen. So kann man dem Lernenden vorrechnen, weshalb seine Lösung nicht korrekt ist. Eine weitere Möglichkeit besteht darin, dass der Lernende die Lösung nicht selbst in das Fill-In Kästchen eingibt, sondern er die Möglichkeit hat aus einer vorgegebenen Drop-Down-Liste die richtige Lösung auszuwählen. Diese Art der Aufgabenstellung entspricht formal einer Single-Choice-Aufgabe, allerdings kann hier bei variabler Aufgabenstellung aufgrund der Auszeichnung durch den oben beschriebenen booleschen Ausdruck auch jeweils eine andere Antwort richtig sein (s. Beispiele). Wie bereits oben für Multiple-Choice-Aufgaben beschrieben, können auch hier Hinweise eingeholt sowie Aufgaben wiederholt werden.

5.2.1 Beispiel: Kurvendiskussion einer Parabel

Abbildung 5.3 zeigt eine Aufgabe, bei der die gegebene Parabel daraufhin zu analysieren ist, ob sie nach oben oder nach unten geöffnet ist und mit der Normalparabel zu verglichen werden muss. Dafür sind zwei Drop-Down-Menüs mit Werten zu belegen. Beim ersten Drop-Down-Menü muss zwischen den Werten *oben* und *unten* gewählt werden und beim zweiten

The screenshot shows a web application interface. At the top left, there is a logo for the Faculty of Business Sciences and the Institute for Informatics and Business Informatics (ICB), along with the name of the professor, Michael Goedicke. The top right corner features the logo of the University of Duisburg-Essen. Below the header, the main content area is titled "Aufgabe Kurvendiskussion Parabel". Underneath, there is a section for "Frage 1" which asks the user to determine the properties of a parabola $f(x) = 17x^2 - 68x - 204$. The question includes a dropdown menu for the opening direction and a text input field for the slope. At the bottom of the main area, there are buttons for "Hinweis", "Überspringen", and "Einreichen". The sidebar on the right contains a "MENÜ" section with options like "Abmelden" and "Hilfe", a "Hauptmenü" with "Übersicht", and language selection options for "English" and "Deutsch". There is also a logo for "PALUNO" and a circular logo for "Bildungsgerechtigkeit im Fokus".

Abbildung 5.3: Ansicht einer Fill-In-Aufgabe für Studierende

```
<task> Wir betrachten die Parabel  $f(x) = \frac{num=[var=z]}{denom=[var=n]} x^2 + \frac{num=[var=u]}{denom=[var=n]} x + \frac{num=[var=v]}{denom=[var=n]}$ . Wir wollen die wichtigsten Eigenschaften des Funktionsgraphen von  $f$  anhand der Funktionsvorschrift bestimmen: Der Graph von  $f$  ist nach  geöffnet und ist  der Graph der Normalparabel.
</task>
```

Listing 5.2: XML-Definition des Drop-Down-Menüs in der Fragestellung einer Aufgabe, Text in eckigen Klammern dient als Platzhalter für Variablen (var=...), Fill-In-Kästchen (fillIn...) bzw. Brüche (frac...). Die Platzhalter werden vor der Anzeige auf einer Webseite vom System ersetzt.

Drop-Down-Menü stehen die Werte *steiler als*, *flacher als* und *genau so steil wie* zur Auswahl. In der zugehörigen XML-Definition kann man dies in Listing 5.2 ablesen.

Es handelt sich bei dieser Aufgabe jedoch um eine Aufgabe mit Variablen, die zu analysierende Parabel ist folglich jedes Mal eine Andere. Genauer gesagt wird bei jeder Anzeige der Aufgabe eine Parabel der Form $y = ax^2 + bx + c$ erstellt, wobei die Parameter a , b und c variabel sind. Dabei kann der Parameter a sowohl positiv, als auch negativ und betragsmäßig kleiner, größer oder gleich 1 sein. Da die richtige Lösung vom Parameter a abhängt, gibt es somit insgesamt sechs mögliche richtige Lösungen, welche vom Lehrenden in der Aufgabenstellung ausgezeichnet werden müssen. Ist z.B. $a > 1$, so muss der Lernende die Werte *oben* und *steiler als* für die richtige Lösung auswählen. Der boolesche Ausdruck, welcher diese Kombination beschreibt ist:

$$a > 1 \text{ UND } [\text{pos}=1] == \text{'oben'} \text{ UND } [\text{pos}=2] == \text{'steiler als'}$$

5 Überblick über Aufgaben und Aufgabentypen

The screenshot shows a web application interface for a task titled "Aufgabe 'Hotel Paluno'". The header includes the logo of the Faculty of Business Sciences and the Institute for Informatics and Business Informatics (ICB), along with the name of Prof. Dr. Michael Goedicke. The main content area contains the task description: "Das Hotel Paluno hat ausschließlich Einzel- und Doppelzimmer. Insgesamt gibt es 29 Zimmer und 51 Betten. Wie viele Zimmer gibt es von jeder Sorte?". Below the text are two input fields for the number of single and double rooms, with a "Hinweis" (Hint) and "Einreichen" (Submit) button. The footer of the task area shows "JACK (Version 2.3.1) | © 2009-2012 Spezifikation von Softwaresystemen". On the right side, there is a navigation menu with options like "Abmelden", "Hilfe", and "Übersicht", and a language selector for "English" and "Deutsch". At the bottom right, there is a logo for "PALUNO The Ruhr Institute for Software Technology" and a circular logo for "Bildungsgerechtigkeit im Fokus".

Abbildung 5.4: Screenshot einer Fill-In-Aufgabe in der Studierendenansicht. Die angezeigten Zahlenwerte sind variable und können sich bei wiederholtem Aufruf der Aufgabe ändern. Die XML-Definition der Aufgabe ist in Listing 5.3 aufgeführt.

Dabei sind “[pos=1]” und “[pos=2]” Platzhalter und werden vom Checker mit den Eingaben des Lernenden aus dem Drop-Down-Menü belegt. Anschließend wird der Ausdruck evaluiert und die Antwort als richtig akzeptiert, falls die Evaluation *wahr* zurück gibt. Genau so kann der Lehrende beim Feedback verfahren, indem er bspw. ein Feedback für den Fall generieren kann, so dass a negativ ist, der Lernende aber der Meinung ist, die Parabel sei nach oben geöffnet.

5.2.2 Beispiel: Lineares Gleichungssystem lösen

Abbildung 5.4 zeigt eine Aufgabe, bei welcher der Lernende ein lineares Gleichungssystem mit zwei Gleichungen und zwei Variablen aufstellen und lösen muss. Die zugehörige XML-Definition ist in Listing 5.3 wiedergegeben. Es stehen diesmal keine Drop-Down-Menüs zur Verfügung, sondern die Lösungen müssen per Tastatur in die beiden dafür vorgesehenen Kästchen eingetragen werden. Auch hier ist die Aufgabestellung variabel, die Zahl der Einzel- und Doppelzimmer wird vor jeder Anzeige vom System neu ausgewürfelt. Da im Kontext der Aufgabenstellung nur ganzzahlige Lösungen sinnvoll sind, werden jedoch zunächst die beiden Lösungen zufällig vom System bestimmt und anschließend die anzuzeigenden Parameter errechnet. Daher werden im letzten Abschnitt der XML-Datei nicht zwei, sondern vier Variablen deklariert. Die ersten beiden enthalten die gewünschte Lösung, während in den letzten beiden die in der Aufgabenstellung angezeigten Zahlen berechnet werden.

Zu dieser Aufgabe wurden neben der korrekten Lösung fünf Feedbackfälle unterschieden und für diese jeweils eine Nachricht generiert:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<exercise>
  <task>
    Das Hotel Paluno hat ausschließlich Einzel- und Doppelzimmer. Insgesamt gibt es  $[\text{var}=\text{n}]$ 
     $[\text{var}=\text{m}]$  Betten. Wie viele Zimmer gibt es von jeder Sorte?

    Das Hotel Paluno hat  $[\text{fillIn groesse}="8"]$  Einzel- und  $[\text{fillIn groesse}="8"]$  Doppelzimmer.
  </task>
  <advice>
    <option> Versuche, ein lineares Gleichungssystem aufzustellen, welches das Problem
    beschreibt. </option>
  </advice>
  <correctanswer>
    <option result="[\text{pos}=1]==[\text{var}=\text{a}]&\&[\text{pos}=2]==[\text{var}=\text{b}] " />
    Richtig. Sehr gut!
  </correctanswer>
  <feedback>
    <option result="[\text{pos}=1]==[\text{var}=\text{b}]&\&[\text{pos}=2]==[\text{var}=\text{a}] " points="0"> Da hast du wohl
    Einzel- und Doppelzimmer vertauscht. Vielleicht ein neuer Versuch? </option>
    <option result="[\text{pos}=1]<0||[\text{pos}=2]<0" > Leider falsch. Die Anzahl der Zimmer kann
    natürlich nicht kleiner als 0 sein. </option>
    <option result="[\text{pos}=1]+[\text{pos}=2]==[\text{var}=\text{n}] " > Leider falsch. Bei  $[\text{pos}=1]$  Einzel- und  $[\text{pos}=2]$ 
    Doppelzimmern gibt es zwar  $[\text{var}=\text{n}]$  Zimmer, aber  $[\text{pos}=1] + 2 \cdot [\text{pos}=2] = [\text{eval}="[\text{pos}=1]+2*([\text{pos}=2])"]$ 
    Betten. </option>
    <option result="[\text{pos}=1]+2*[\text{pos}=2]==[\text{var}=\text{m}] " > Leider falsch. Bei  $[\text{pos}=1]$  Einzel- und
     $[\text{pos}=2]$  Doppelzimmern gibt es zwar  $[\text{var}=\text{m}]$  Betten, aber  $[\text{pos}=1] + [\text{pos}=2] = [\text{eval}="[\text{pos}=1]+[\text{pos}=2]"]$ 
    Zimmer. </option>
    <option result="andere" > Leider falsch. Weder die Anzahl der Zimmer noch die Anzahl
    der Betten stimmt bei diesen Angaben mit den Vorgaben überein. </option>
  </feedback>
  <variables>
    <option name="a" min="2" max="25" />
    <option name="b" min="2" max="40" />
    <option name="n" value="[\text{var}=\text{a}]+[\text{var}=\text{b}] " />
    <option name="m" value="[\text{var}=\text{a}]+2*[\text{var}=\text{b}] " />
  </variables>
</exercise>
```

Listing 5.3: XML-Definition der Fill-In-Aufgabe aus Abbildung 5.4.

1. Vertauscht der Lernende in seiner Rechnung die Variablen für Einzel- und Doppelzimmer, gibt also den richtigen Wert für die Einzelzimmer in das Feld für die Doppelzimmer ein und umgekehrt, so weist das System ihn auf diesen Umstand hin.
2. Ist eine der beiden Lösungen kleiner als 0, so wird der Lernende darauf hingewiesen, dass dies nicht möglich ist.
3. Stimmt die Anzahl der Betten, ist jedoch die Anzahl der Zimmer falsch, so wird dem Lernenden die Anzahl der Zimmer bei seiner Lösung vorgerechnet und mit der geforderten Anzahl verglichen.
4. Stimmt die Anzahl der Zimmer, aber nicht die Anzahl der Betten, so wird dem Lernenden die Anzahl der Betten bei seiner Lösung vorgerechnet und mit der geforderten Anzahl verglichen. Die Ausgabe dieses Feedbacks ist in Abbildung 5.5 dargestellt.
5. Stimmen weder die Anzahl der Betten, noch die Anzahl der Zimmer, so wird der Lernende auf diese Tatsache hingewiesen.

5 Überblick über Aufgaben und Aufgabentypen

The screenshot shows a web page for a task titled "Aufgabe 'Hotel Paluno'". At the top left, there is a logo of a stylized person and text: "Fakultät für Wirtschaftswissenschaften >> Institut für Informatik und Wirtschaftsinformatik (ICB) >> Spezifikation von Softwaresystemen | Prof. Dr. Michael Goedicke". At the top right, the logo of "UNIVERSITÄT DUISBURG ESSEN" is visible, along with a "MENÜ" button and a user login status "[Eingeloggt als ...]". Below the menu are buttons for "Abmelden" and "Hilfe". A "Hauptmenü" section contains a "Übersicht" button and language options for "English" and "Deutsch". The main content area has a title "Aufgabe 'Hotel Paluno'" and a question: "Das Hotel Paluno hat ausschließlich Einzel- und Doppelzimmer. Insgesamt gibt es 39 Zimmer und 54 Betten. Wie viele Zimmer gibt es von jeder Sorte?". Below the question are two input fields: "Das Hotel Paluno hat Einzel- und Doppelzimmer." Below the input fields are buttons for "Hinweis" and "Einreichen". A "Feedback:" section shows the message: "Leider falsch. Bei 20 Einzel- und 19 Doppelzimmern gibt es zwar 39 Zimmer, aber $20 + 2 \cdot 19 = 58$ Betten." Below the feedback is an "Ergebnisübersicht" section showing "Gesamtergebnis: 0" with a sad face icon. A note below states: "Gesamtergebnis ist berechnet als InstantFillInChecker (1) result. Mindestergebnis für eine korrekte Lösung ist 50". On the right side of the page, there is a logo for "PALUNO The Ruhr Institute for Software Technology" and a circular logo with the text "Bildungsgerechtigkeit im Fokus". At the bottom right, there is a logo for the "Bundesministerium für Bildung und Forschung".

Abbildung 5.5: Ansicht des Feedbacks nach einer Eingabe zu der bereits in Abbildung 5.4 gezeigten Fill-In-Aufgabe.

5.3 Mehrstufige Aufgaben

Mehrstufige Aufgaben ermöglichen es dem Lehrenden, beliebig viele Einzelaufgaben vom Typ Multiple-Choice oder Fill-In zu kombinieren und den Lernenden so in Schritten zur korrekten Lösung zu führen. Dem Lernenden werden dabei immer nur die bereits bearbeiteten Stufen angezeigt. Bearbeitet er eine Stufe korrekt, so erscheint neben dem Feedback auch sofort die nächste zu bearbeitende Stufe. Unterläuft ihm bei der Bearbeitung ein Fehler, so muss er die Stufe erneut bearbeiten. Wenn der Lernende eine Frage nicht beantworten kann oder möchte, so kann er diese mittels des dafür vorgesehenen Buttons überspringen und bekommt sowohl die korrekte Antwort als auch die nächste Stufe angezeigt. Hat der Lernende alle Stufen korrekt bearbeitet oder übersprungen, so ist die Aufgabe beendet und die Buttons zum Einreichen einer Lösung, Überspringen und zum Einholen eines Hinweises werden deaktiviert. Der Lehrende kann für jede Stufe eine Gewichtung in Form einer Dezimalzahl zwischen 0 und 1 definieren, welche den Einfluss des Resultats der Stufe auf die Gesamtpunktzahl festlegt. Dabei ist darauf zu achten, dass die Summe der Gewichtungen immer 1 ergibt. Der Lernende wird durch eine Tabelle unter der Aufgabe darüber informiert, wie viele Stufen die Aufgabe besitzt und wie die einzelnen Stufen gewichtet werden. Zudem kann er sich in der Tabelle über seinen aktuellen Gesamtpunktstand und seine Resultate bei den einzelnen Stufen informieren.

Aufgabe "Kurvendiskussion einer Parabel"

Frage 1

Wir betrachten die Parabel $f(x) = \frac{17}{10}x^2 - \frac{34}{5}x - \frac{17}{2}$. Wir wollen die wichtigsten Eigenschaften des Funktionsgraphen von f anhand der Funktionsvorschrift bestimmen:

Der Graph von f ist nach geöffnet und ist der Graph der Normalparabel.

Punkte: 100/100



Feedback:

Richtig. Sehr schön.

Frage 2

Welcher der folgenden Punkte ist der Punkt in welchem der Graph die y -Achse schneidet?

Antworten:

- $(0, -\frac{17}{2})$
- $(1, -\frac{68}{5})$
- $(0, \frac{17}{10})$
- $(-6, 0)$

[Hinweis](#) [Überspringen](#) [Einreichen](#)

Ergebnisübersicht

Frage	Punkte	Gewichtung
1	100	0.2
2	noch nicht bearbeitet	0.1
3	noch nicht bearbeitet	0.4
4	noch nicht bearbeitet	0.3
Gesamt	20	

Abbildung 5.6: Ausschnitt einer mehrstufigen Aufgabe. Es werden die ersten zwei von insgesamt vier Stufen angezeigt.

5 Überblick über Aufgaben und Aufgabentypen

Mehrstufige Aufgaben eignen sich besonders, um bei Aufgaben mit langem Rechenweg Zwischenresultate abzu prüfen. So können Fehler exakt lokalisiert werden und es kann im Feedback genau auf diese eingegangen werden. Außerdem lässt sich so verhindern, dass der Lernende mit einem falschen Ergebnis weiterrechnet, wodurch Folgefehler entstehen können. Des Weiteren lassen sich mit mehrstufigen Aufgaben verschiedene Resultate zu einem Thema zusammenfassen, wie etwa bei einer Kurvendiskussion. Die in Abschnitt 5.2.1 besprochene Fill-In-Aufgabe ist beispielsweise Teil einer mehrstufigen Aufgabe. Ihre Fortsetzung mit der zweiten aktiven Stufe ist im Screenshot in Abbildung 5.6 dargestellt.

Bei dieser Aufgabe ist eine Parabel gegeben, welche vom Lernenden so analysiert werden soll, dass er anschließend in der Lage wäre, die Parabel zu zeichnen. Wie bereits besprochen, sollen in Stufe 1 zunächst anhand der Funktionsgleichung die Richtung der Öffnung der Parabel bestimmt und ein Vergleich zur Normalparabel gezogen werden. Stufe 2 ist eine Multiple-Choice-Aufgabe, in welcher der y -Achsenabschnitt der Parabel bestimmt werden soll. Als Distraktoren dienen ein weiterer Punkt mit x -Wert 0, ein Punkt mit y -Wert 0 sowie ein Punkt auf der Parabel, welcher weder einer der Nullstellen, noch dem y -Achsenabschnitt entspricht. In Stufe 3 sollen in einer Fill-In-Aufgabe die Nullstellen der Parabel berechnet werden. Verrechnet sich der Lernende hierbei, so wird ihm im Feedback vorgerechnet, welche y -Werte man erhält, wenn man die von ihm eingegebenen x -Werte in die Parabelgleichung einsetzt. Stufe 4 ist ebenfalls eine Fill-In-Aufgabe, in welcher der Scheitelpunkt der Parabel bestimmt werden soll. Mit der Bearbeitung dieser Stufe ist die Aufgabe beendet, denn der Lernende hat nun alle Informationen, um die Parabel in ein Koordinatensystem zu zeichnen.

6 Erfahrungen aus dem Vorkurs

Im Sommer 2012 ergab sich die Möglichkeit, den neu programmierten Teil von JACK in einem Mathematik-Vorkurs für die Fakultät Wirtschaftswissenschaften zu testen. An diesem Vorkurs nahmen 150 StudienanfängerInnen der Studiengänge „Betriebswirtschaftslehre“, „Volkswirtschaftslehre“ und „Angewandte Informatik - Systems Engineering“ teil. Im Folgenden wird kurz das didaktische Konzept des Vorkurses vorgestellt. Anschließend werden einige Erfahrungen diskutiert, die als erste Evaluation der bisherigen Projektergebnisse genutzt werden können.

6.1 Aufbau und Ziele des Vorkurses

Der Vorkurs erstreckte sich über drei Wochen im September, mit insgesamt zwölf Präsenzterminen in Form einer zweistündigen Übung. Jeder Übung wurde ein Themengebiet zugeteilt und zu jedem Themengebiet gab es ein Übungsblatt für die Übung, welches zunächst von den StudienanfängerInnen eigenständig bearbeitet und dann mit den Tutoren in der Gruppe besprochen wurde. Unabhängig von diesem Übungsblatt gab es pro Themengebiet zehn bis zwanzig Aufgaben in JACK, die Tag für Tag schrittweise veröffentlicht wurden. Die StudienanfängerInnen sollten die Aufgaben in JACK zur Vorbereitung und Nachbereitung der Übung bearbeiten, so dass in der Mischung aus Präsenzübung und E-Learning-Elementen ein typisches Blended-Learning-Szenario realisiert werden konnte. Wenn die StudienanfängerInnen Fragen zu JACK hatten, konnten sie diese jederzeit in den Übungen stellen. Es sollte somit sichergestellt sein, dass die StudienanfängerInnen nicht mit dem System JACK allein gelassen werden.

Das Ziel des Vorkurses war, eine Brücke zwischen der Schule und dem Studium zu schaffen. Die StudienanfängerInnen sollten nach diesem Vorkurs das mathematische Handwerkszeug beherrschen, sprich Termumformungen, Systeme linearer Gleichungen lösen, mit Potenzen und Wurzeln rechnen können, Ungleichungen lösen können, usw., welches in der Schule bis zur Klasse 10 gelehrt wird. Themen wie Differential- und Integralrechnung wurden außen vor gelassen, so wie auch die Wahrscheinlichkeitstheorie und Statistik. Es gab also keinen neuen Stoff. Dadurch war es den StudienanfängerInnen möglich, die Aufgaben in JACK zur Vorbereitung der Übung ohne genauere Erklärungen und Einführungen in das jeweilige Thema lösen zu können.

Am Ende des Vorkurses wurde eine Klausur in JACK gestellt. Die Klausur war 4 Stunden freigeschaltet, aber jede/r StudienanfängerIn durfte sie nur einmal bearbeiten und einreichen. Dadurch sollte eine typische Prüfungssituation geschaffen werden. Die StudienanfängerInnen hatten zwar die Möglichkeit diesen Test zu Hause zu bearbeiten, sollten sich aber trotzdem genaue Gedanken über ihre Lösungen machen, da es keine zweite Möglichkeit der Einreichung gab. Insgesamt wurden 10 Aufgaben in der Klausur gestellt, wobei in jeder Aufgabe maximal 100 Punkte erreicht werden konnten. Die Klausur galt bei einer Punktezahl von 500 als bestanden. Es reichte also 50 % der Aufgaben richtig zu beantworten. Um allen Teilnehmern eine gleiche Chance zu geben, wurden anschließend alle eingereichten Lösungen noch zusätzlich von Hand korrigiert. Somit hatte kein/e StudienanfängerIn einen Nachteil, falls das System eine eingereichte, korrekte Lösung nicht erkannt hatte. Diese Klausur wurde von gut 17 % der Vorkursteilnehmer bearbeitet, von denen 65 % die Klausur bestanden. Diejenigen, die die Klausur nicht bestanden, hatten zum größten Teil nicht alle Aufgaben bearbeitet. Meistens wurde die Klausur auch nur knapp nicht bestanden, d.h. die StudienanfängerInnen beherrschten den Stoff, der in dem Vorkurs gelehrt wurde.

6.2 Aufgabenerstellung und Qualitätssicherung

Für den Vorkurs wurden Aufgaben zu den zwölf Themengebieten „Terme mit einer Variablen“, „Terme mit mehreren Variablen“, „Lineare Funktion“, „Systeme linearer Gleichungen“, „Bruchterme“, „Quadratwurzel“, „Quadratische Funktion und ihr Graph“, „Quadratische Gleichungen“, „Flächensätze am Dreieck“, „Potenzen“, „Exponentialfunktionen“ und „Trigonometrie“ entwickelt. Da in diesem Vorkurs reiner Schulstoff wiederholt wurde, wurde auf eine ausführliche Einführung und Erklärung im System verzichtet. Bei der Aufgabenstellung wurden Lehrbücher aus der Schule zu Hilfe genommen. Dadurch sollte gewährleistet sein, dass die Aufgaben so ähnlich gestellt waren, wie StudienanfängerInnen dies aus der Schule gewohnt sind. Dabei wurde davon ausgegangen, dass die StudienanfängerInnen bei Problemen in ihren Unterlagen aus der Schule nachschlagen, im Internet nachschauen oder in den Übungen nachfragen konnten. Außerdem sollte bei einer Falscheinreichung das Feedback die StudienanfängerInnen zur richtigen Lösung lotsen. Zudem sollte auch der „Hinweis“-Button als Hilfe bei der Lösung der Aufgaben benutzt werden und somit ein selbstständiges Lernen möglich sein.

Für den Vorkurs wurden insgesamt ca. 200 Aufgaben zu den oben genannten Themengebieten von zwei wissenschaftlichen Mitarbeitern und drei studentischen Hilfskräften erstellt. Diese arbeiteten über einen Zeitraum von 1,5 Monaten wöchentlich ca. 70 Arbeitsstunden an den Aufgaben. Die Studentischen Hilfskräfte waren zu Beginn noch nicht mit dem System vertraut und mussten daher zunächst eingearbeitet werden, was den Arbeitsfluss zunächst deutlich verlangsamte. Wenn möglich wurde darauf geachtet, die Aufgabenstellung variabel

Erster Jahresbericht zum Projekt "Bildungsgerechtigkeit im Fokus" (Teilprojekt 1.2 - "Blended Learning") an der Fakultät für Wirtschaftswissenschaften

zu gestalten, was jedoch zusätzlichen Arbeitsaufwand bedeutete, da es einige Übung erfordert, die Variablen so zu belegen, dass die Lösungen "schön", d.h. zum Beispiel stets ganzzahlig, sind. Bei der Nullstellenbestimmung bei einer Parabel ist es bspw. nicht zielführend, drei Variablen a, b und c auszuwürfeln und dem Lernenden die Parabel $y = ax^2 + bx + c$ zu präsentieren. Zum einen kann es nämlich passieren, dass die Parabel keine Nullstellen in den reellen Zahlen besitzt, zum anderen könnten die Nullstellen Dezimalzahlen mit vielen Nachkommastellen sein, was aus didaktischer Sicht nicht geeignet wäre. Es ist also notwendig, die Aufgabe und die aus den obigen Überlegungen entstehenden Bedingungen an die Aufgabe genau zu analysieren, um die Variablenbelegung sinnvoll zu gestalten. Das bedeutet natürlich einen erheblichen zeitlichen Mehraufwand, welcher jedoch dadurch wettgemacht wird, wenn die gleiche Aufgabe nicht wieder und wieder gestellt werden muss.

Zur Qualitätssicherung der Aufgaben nach der Durchführung des Vorkurses wurden alle Lösungen aller StudienanfängerInnen durchgesehen. Dabei wurde festgestellt, dass Einreichungen bei Fill-In-Aufgaben vor allem aus den folgenden Gründen als nicht korrekt gewertet wurden: Neben tatsächlichen (Rechen-)Fehlern kamen mathematisch nicht korrekte Schreibweisen vor, bei denen z.B. anstatt einer durch die Aufgabenstellung vorgegebenen Variable die Variable x geschrieben wurde, sowie syntaktive Fehler, bei denen z.B. statt $*$ das \times als Multiplikationszeichen verwendet wurde. Durch die penible, negative Reaktion des Systems auf solche Fehler sollten die Studierenden unter anderem lernen, wie wichtig es ist, Lösungen mathematisch korrekt zu schreiben und auch die richtigen Variablen zu benutzen. Viele StudienanfängerInnen haben damit Schwierigkeiten, da ihre Schreibweise in der Schule nie so penibel bewertet wurde. Sie sehen ihre Fehler diesbezüglich nicht immer ein, sondern geben dem System die Schuld.

Bei der Qualitätssicherung ist aufgefallen, dass die Erstellung von Fill-In-Aufgaben erheblich schwieriger ist, als die Erstellung von Multiple-Choice-Aufgaben. Der Vorteil von Multiple-Choice-Aufgaben liegt darin, dass die Antwortmöglichkeiten bereits vorgegeben sind, sie sind sogenannte geschlossene Aufgaben. Dadurch fällt es leichter, zu den jeweiligen Antwortmöglichkeiten Feedback zu schreiben. Bei Fill-In-Aufgaben sieht dies etwas anders aus. Hier muss man sich zunächst über alle möglichen Fehler der Studierenden Gedanken machen. Nur so kann das System dem Studierenden auch bei der Findung der richtigen Lösung helfen. In Zukunft soll mehr Zeit für die Erstellung von Fill-In-Aufgaben investiert werden. Eine große Anzahl von eingereichten Lösungen kann dem Lehrenden helfen, eventuelle Fehlermöglichkeiten ausfindig zu machen und für diese Feedbackfälle zu generieren. Dies bedeutet, dass auch in Zukunft viel Zeit in die Qualitätssicherung fließen wird.

6.3 Evaluation

Leider wurde JACK nicht so intensiv genutzt, wie dies gewünscht war. Während zu Beginn des Vorkurses ca. 130 der 150 StudienanfängerInnen das System ausprobierten, schrumpfte die Zahl der NutzerInnen pro Tag im weiteren Verlauf auf unter 5. Bei einigen StudienanfängerInnen kam Frust auf, da ihre Einreichungen als „falsch“ bewertet wurden, obwohl sie der Meinung waren ihre Antwort wäre richtig. Tatsächlich bewertete das System die Antworten aber in den meisten Fällen zutreffenderweise als falsch. Auch wenn die StudienanfängerInnen demnach selten an einem tatsächlichen Systemfehler gescheitert waren, gaben sie schnell auf und betrachteten JACK als fehlerhaft, anstatt über die Richtigkeit der eigenen Antwort nachzudenken. Als erste Erfahrung kann damit festgehalten werden, dass eindeutiges und motivierendes Feedback sehr wichtig ist, wenn NutzerInnen nicht abgeschreckt werden sollen. Auf Nachfrage der Tutoren in den Übungen, warum das System nicht häufiger benutzt wurde, kamen ansonsten noch die Antworten, dass man keine Zeit habe oder die Präsenzveranstaltung völlig ausreichen würde. Es kann somit zudem vermutet werden, dass sich die Teilnehmer eines sehr frei gestalteten Blended-Learning-Szenarios wie dieses Vorkurses, die Bestandteile herausuchen, die ihren persönlichen Bedarf am effektivsten decken. Da in Zukunft jedoch für die zu erwartende Zahl von TeilnehmerInnen ein solch flexibles System nicht angeboten werden kann, wird eine Fokussierung auf E-Learning und zusätzliche Unterstützung gesetzt, jedoch keine separaten Kurse angeboten werden.

6.3.1 Evaluationsbögen

Um einen weiteren Einblick zu bekommen, wie die Teilnehmer das Arbeiten mit dem System JACK fanden und auch um herauszufinden, warum nicht häufiger Aufgaben in JACK gelöst worden waren, wurden in der letzten Übungsstunde Evaluationsbögen verteilt. Dort sollten die StudienanfängerInnen angeben, wie sie die Zusammenarbeit mit dem System und dem Vorkurs fanden. Von den 150 Teilnehmern kamen am Ende 88 Evaluationsbögen zurück, so dass davon auszugehen ist, dass die Meinung in den Evaluationsbögen die allgemeine Meinung widerspiegelt. Ein Teil der Antworten wird in den Abbildungen 6.1 bis 6.6 grafisch wiedergegeben.

Zunächst sollte evaluiert werden, ob der Vorkurs überhaupt dem Bedarf entsprach. Wie man in Abbildung 6.1 ablesen kann fanden 69 Teilnehmer das Niveau des Vorkurses „genau richtig“, d.h. sie fanden die Aufgaben weder zu schwer noch zu leicht. Auch die Aussage in Abbildung 6.2 von 70 Teilnehmern, das „Themenspektrum des Vorkurses war genau richtig“, unterstreicht den bedarfsgerechten Aufbau des Vorkurses. Außerdem sollten die Studierenden den Vorkurs (s. Abbildung 6.3) auch allgemein beurteilen. Dabei empfanden 51 Studierende den Vorkurs als „eher hilfreich“ und 32 sogar „sehr hilfreich“.

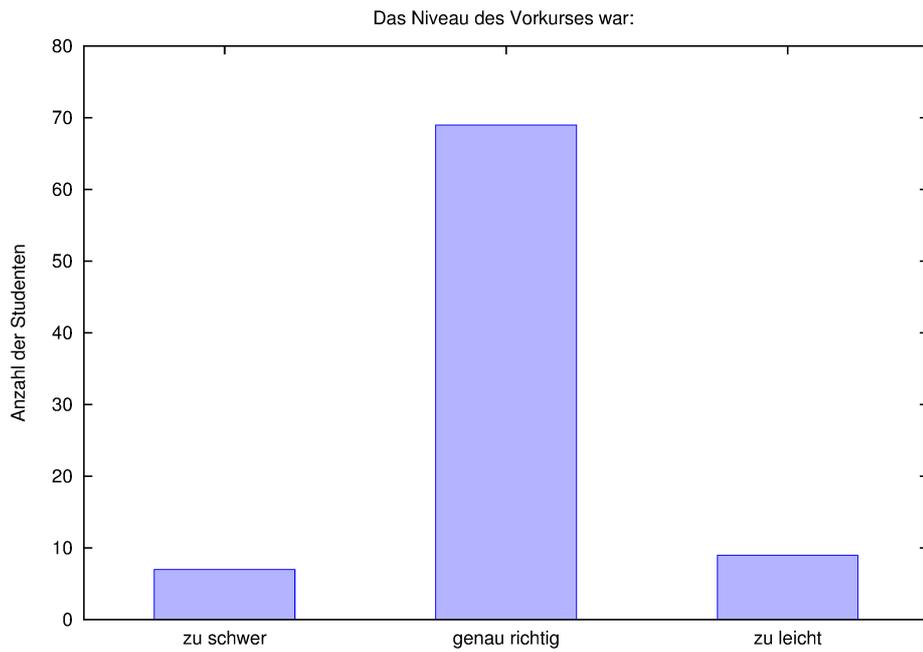


Abbildung 6.1: Die Studierenden sollten das Niveau des Vorkurses beurteilen.

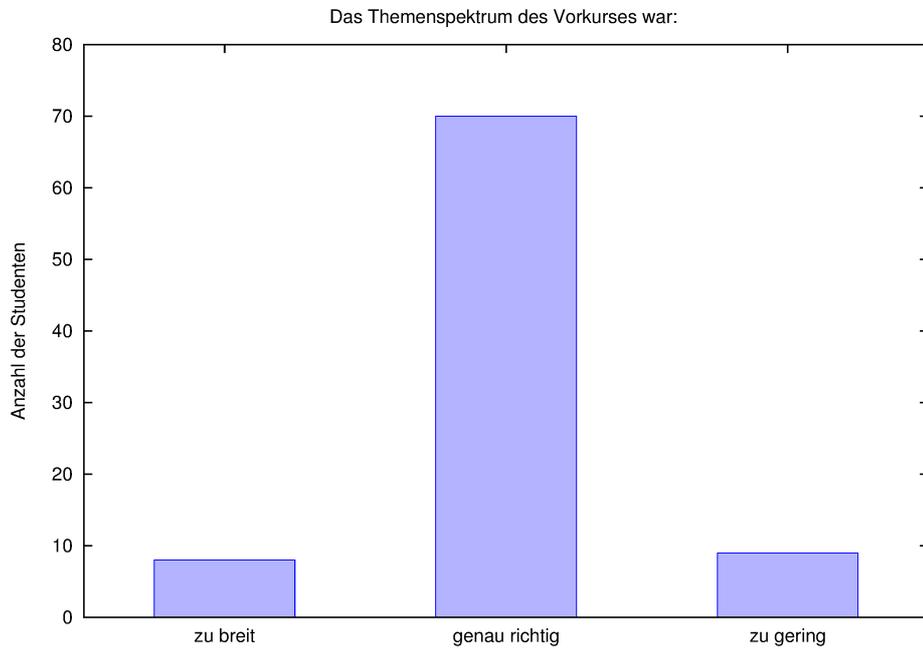


Abbildung 6.2: Die Studierenden sollten das Themenspektrum des Vorkurses beurteilen.

6 Erfahrungen aus dem Vorkurs

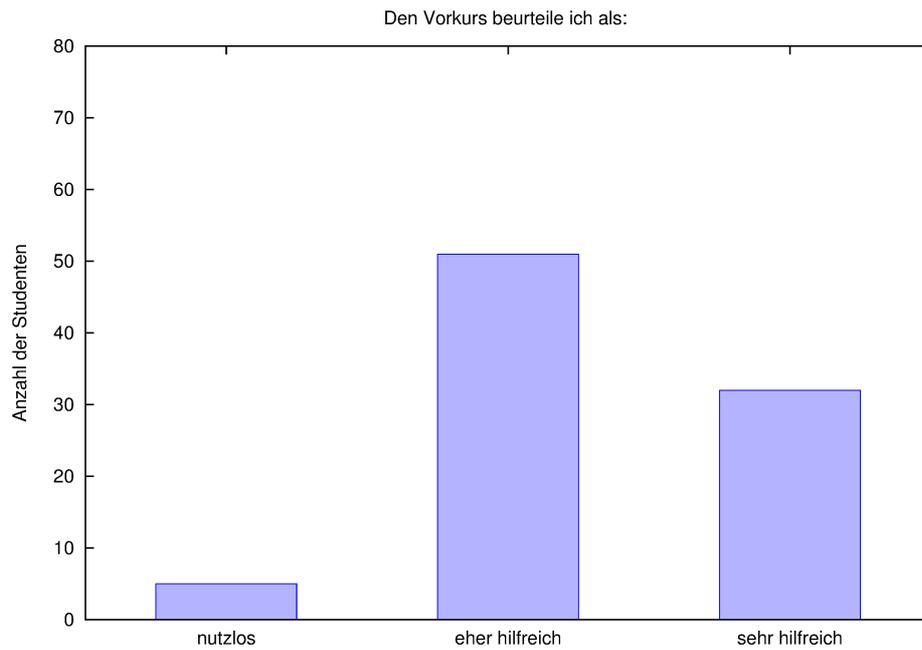


Abbildung 6.3: Die Studierenden sollten den Vorkurs allgemein beurteilen.

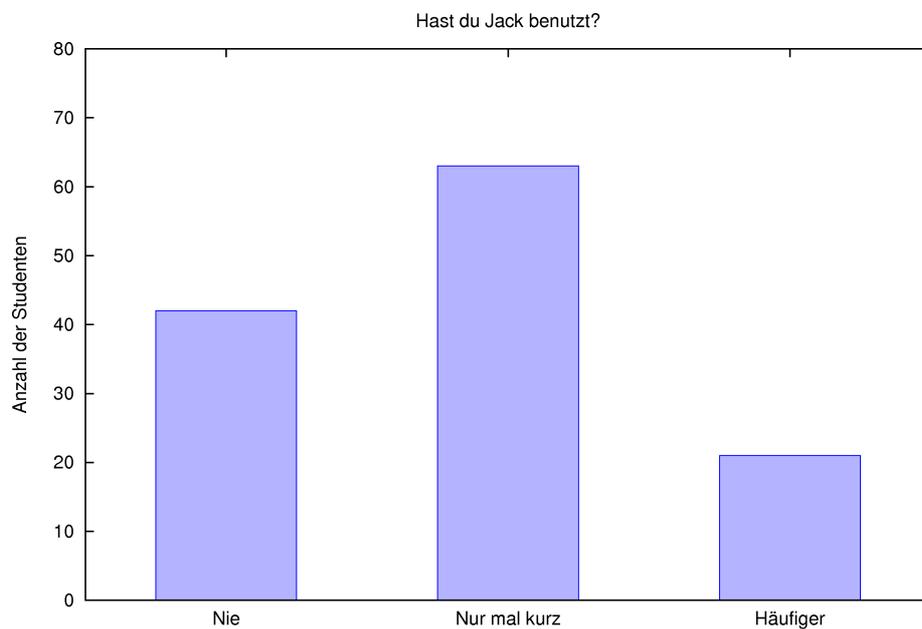


Abbildung 6.4: Die Studierenden sollten angeben, wie häufig sie das System JACK während des Vorkurses benutzt haben. Dabei ist zu beachten, dass einige Studierende mehrere Sachen angekreuzt haben, so dass es am Ende mehr als 88 Antworten gab.

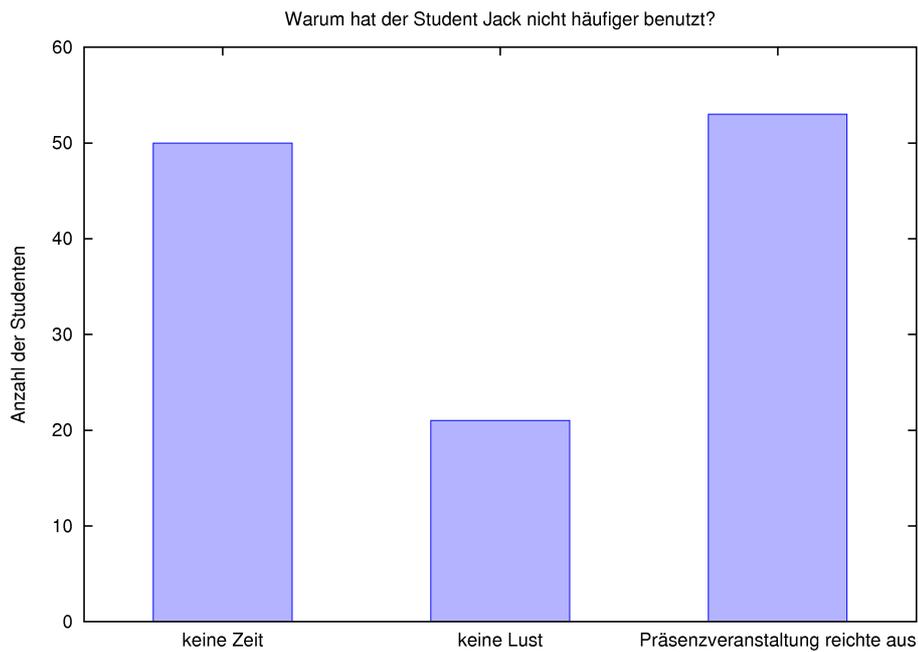


Abbildung 6.5: Die Studierenden sollten angeben, warum sie das System JACK nicht häufiger während des Vorkurses benutzt haben.

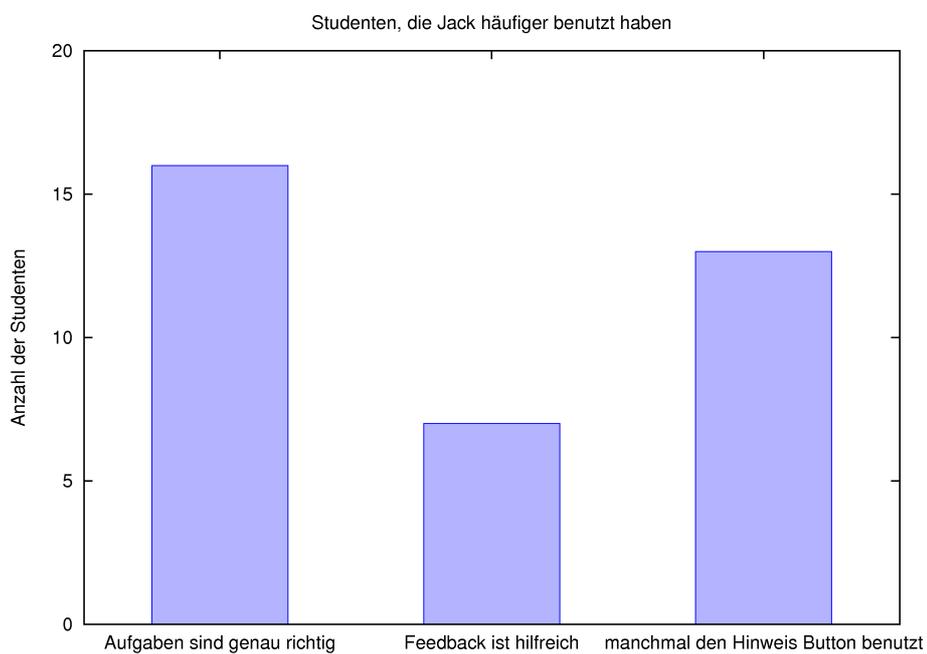


Abbildung 6.6: Es haben 21 Studierende angegeben das System JACK öfter benutzt zu haben. Diese hatten die folgende Meinung über Aufgaben, Feedback und Hinweise

Da JACK nicht von jedem Teilnehmer des Vorkurses benutzt worden war, wurde versucht, mit den Evaluationsbögen den Grund für dieses Verhalten herauszufinden. Dazu wurde zunächst die Frage gestellt „Hast du JACK benutzt?“ (siehe Abbildung 6.4). Die StudienanfängerInnen hatten drei Antwortoptionen: „Nie“, „Nur mal kurz“ und „Häufiger“. Von den 88 Teilnehmern der Umfrage benutzten laut Evaluationsbogen 42 Teilnehmer JACK nie, 63 nur mal kurz und 21 häufiger, wobei zu berücksichtigen ist, dass 29 Teilnehmer mehr als eine der Antwortmöglichkeiten ankreuzten. Es stellt sich daher die Frage, woran die verhältnismäßig geringe Nutzung von JACK lag. Die meisten Umfrageteilnehmer haben als Grund dafür „keine Lust“, „keine Zeit“ und „mir hat die Präsenzveranstaltung ausgereicht“ angegeben (siehe Abbildung 6.5). Bei der Interpretation ist zu bedenken, dass der Vorkurs freiwillig war und die StudienanfängerInnen am Ende keinen Schein bekamen. Womöglich wollten sie nicht ihre ganze Freizeit vor Beginn des Studiums dem Vorkurs opfern. Einige Teilnehmer gaben aber auch technische Schwierigkeiten als Grund an. Diese Aussage wird sehr ernst genommen und es wird daran gearbeitet, alle Fehler im System zu beheben. Weiterhin wird daran gearbeitet, dass das System benutzerfreundlicher wird. Was sagen nun die 21 StudienanfängerInnen, die JACK häufiger benutzten? In Abbildung 6.6 kann man ablesen, dass 16 StudienanfängerInnen die Aufgaben als genau richtig einstufen. Außerdem gaben 13 StudienanfängerInnen an, dass sie manchmal den Hinweis Button benutzten und sich somit Hilfe bei der Beantwortung der Fragen holten. Zuletzt sagten 7 StudienanfängerInnen, dass das Feedback ihnen beim Lösen der Aufgaben weiterhalf. In Zukunft wird auch weiterhin viel Zeit und Konzentration in die Erstellung der Hinweise und des Feedbacks investiert. Diese beiden Faktoren sind die wichtigsten bei der Aufgabenstellung. Durch sie ist es (und soll es) den StudienanfängerInnen möglich sein, die Aufgaben selbstständig zu lösen.

6.3.2 Nutzerverhalten

Nach dem Vorkurs wurde das Nutzerverhalten der StudienanfängerInnen in JACK betrachtet. Dabei wurde nur das erste Themengebiet „Terme mit einer Variable“ beurteilt, da hier die meisten Einreichungen gemacht wurden. Dabei sind ein paar Details aufgefallen. Wie man in Abbildung 6.7 gut erkennen kann, nimmt die erreichte Gesamtpunktzahl stetig mit der Anzahl der gelösten Aufgaben zu. Wenn man von allen StudienanfängerInnen, die im ersten Themengebiet eine Einreichung machten, den Mittelwert der erreichten Punkte berechnet (siehe dazu Abbildung 6.8), kann man auch erkennen, dass die StudienanfängerInnen, die viele Aufgaben gelöst hatten, im Schnitt einen höheren Mittelwert erreichten, als die StudienanfängerInnen, die nur sehr wenige Aufgaben gelöst hatten. Dieses Ergebnis kann mehrere Gründe haben. Zum einen werden vermutlich die StudienanfängerInnen, die mit den Aufgaben besser zurecht kamen, auch mehr Aufgaben lösen, als die, die schon mit der ersten Aufgabe große Probleme haben. In beiden Abbildungen kann man z.B. zwei StudienanfängerInnen erkennen, die eine oder zwei Aufgaben gelöst hatten, insgesamt aber keinen einzigen

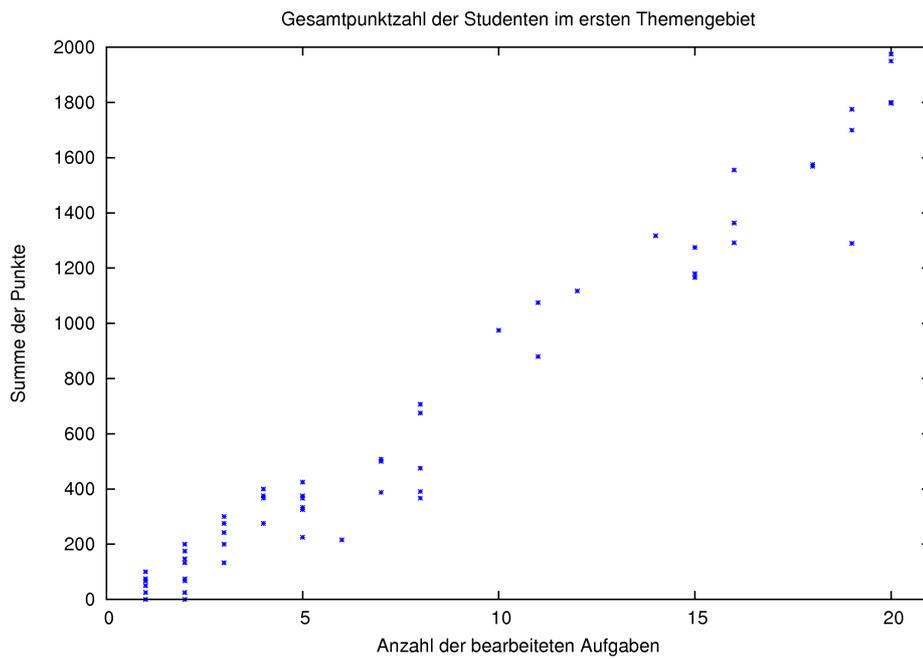


Abbildung 6.7: Statistik über die 20 Aufgaben des ersten Themengebiets. Es werden die Gesamtpunktzahl der Studierenden bezüglich der Anzahl der bearbeiteten Aufgaben beschrieben.

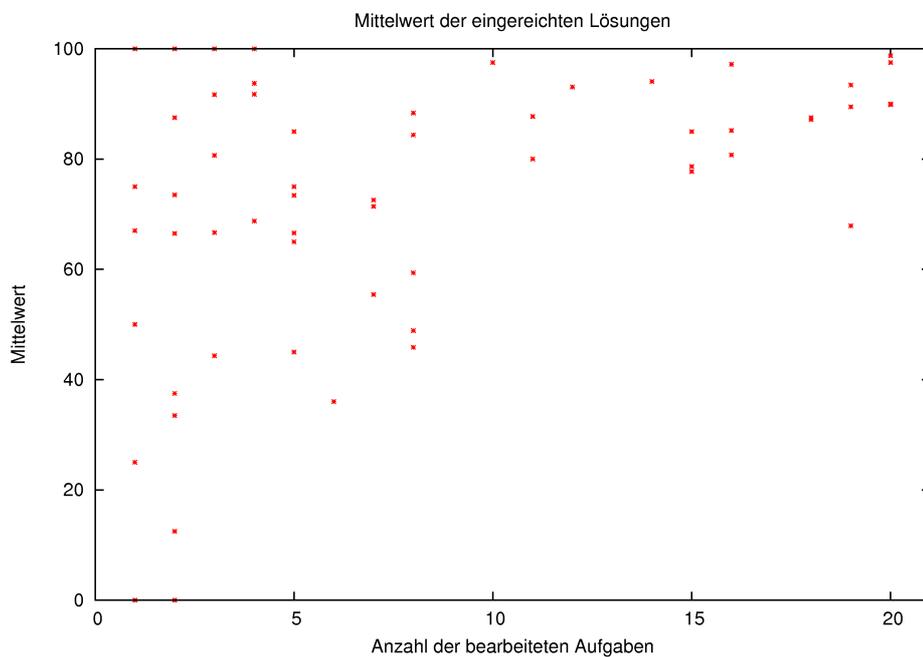


Abbildung 6.8: Der Mittelwert der eingereichten Lösungen des ersten Themengebiets. Insgesamt haben 78 Studierenden Lösungen eingereicht.

6 Erfahrungen aus dem Vorkurs

Aufgaben	Anzahl der StudienanfängerInnen	Anzahl aller Lösungen	Anzahl richtige Lösungen	Anzahl falscher Lösungen	Anzahl der StudienanfängerInnen mit mehr als einer Lösung	Mehrfachversuche in Prozent	Aufgabentyp	Anzahl der StudienanfängerInnen, die mindestens einen Schritt übersprungen haben
Gesetze	64	184	134	50	30	46,88%	Fill-In	-
Löse die Betragsgleichung	49	83	37	38	30	61,22%	Mehrstufig	35
Löse die Ungleichung	19	65	11	54	12	63,16%	Fill-In	-
Löse die Ungleichung 1	11	28	7	21	6	54,55%	Fill-In	-
Lösungsmenge Gleichung	31	46	42	6	3	9,68 %	Fill-In	-
Oberfläche eines Würfels	31	39	33	6	15	48,39%	Mehrstufig	16
Term vereinfachen und Gleichung lösen	30	45	37	8	4	13,33%	Fill-In	-
Term vereinfachen und Gleichung lösen 2	30	35	35	0	11	36,67%	Mehrstufig	5
Terme	32	40	40	0	0	0%	Fill-In	-
Terme aufstellen	26	87	18	69	20	76,92%	Fill-In	-
Terme aufstellen und vereinfachen	21	24	20	4	10	47,62%	Mehrstufig	9
Terme vereinfachen	25	31	26	5	2	8%	Fill-In	-
Terme zusammenfassen	22	32	21	11	5	22,73%	Fill-In	-
Termumformung	23	30	24	6	3	13,04%	Fill-In	-
Termumformungen mit Fehlern	20	58	20	38	15	75%	MC	-
Textaufgabe Terme	20	21	19	2	10	50%	Mehrstufig	6
Ungleichung aufstellen und lösen	11	12	12	0	3	27,27%	Mehrstufig	3
Waagschale	17	19	19	0	3	17,65%	Mehrstufig	1
Wasserhöhe im Swimmingpool	21	26	15	11	8	38,10%	Mehrstufig	9
Zahlenrätsel	22	25	24	1	0	0%	Mehrstufig	6

Tabelle 6.1: Überblick über das Nutzerverhalten im ersten Themenblock des Mathematik-Vorkurses im WS 2012/13.

Punkt bekamen. Ein weiterer Grund könnte aber auch sein, dass die StudienanfängerInnen, die mehr Aufgaben einreichten, mehr geübt hatten und dadurch am Ende einen höheren Mittelwert erreichen konnten. Wenn man sich die beiden Abbildungen genau anschaut, sieht man, dass alle StudienanfängerInnen, die zwanzig Aufgaben gelöst hatten auch im Mittel über 80 Punkte erreichten, also zu über 80% die Fragen richtig beantworteten.

In Tabelle 6.1 kann man ablesen, wie häufig die StudienanfängerInnen versucht hatten, nach

einer falschen eingereichten Lösung die richtige Lösung zu finden. Hier wird wieder nur das erste Themengebiet „Terme mit einer Variable“ betrachtet. In der ersten Spalte steht der Name der jeweiligen Aufgabe, in der zweiten kann man die Anzahl der StudienanfängerInnen ablesen, die diese Aufgabe bearbeiteten, in der dritten Spalte sieht man die Anzahl der eingereichten Lösungen und in der vierten und fünften Spalte kann man die Anzahl der richtigen Lösungen und die Anzahl der falschen Lösungen ablesen. In der sechsten Spalte steht die Anzahl der StudienanfängerInnen, die versucht hatten die richtige Lösung zu finden, dabei müssen sie nicht unbedingt erfolgreich gewesen sein. Hier werden nur die Mehrfacheinreichungen betrachtet, die inhaltlich von den vorherigen Einreichungen derselben Person abweichen. In der siebten Spalte steht diese Angabe in Prozent hinsichtlich aller StudienanfängerInnen der jeweiligen Aufgabe und in der achten steht der Aufgabentyp. Die letzte Spalte gibt bei mehrstufigen Aufgaben an, wie viele StudienanfängerInnen mindestens ein Schritt übersprungen hatten.

Wenn man davon ausgeht, dass die Aufgaben von den StudienanfängerInnen von oben nach unten bearbeitet wurden, wird man feststellen, dass sie am Anfang hoch motiviert waren und versuchten, die richtige Lösung zu finden. Bei den ersten vier Aufgaben haben über 50% versucht, die richtige Lösung zu finden. Eine weitere Beobachtung ist, dass der Aufgabentyp ausschlaggebend dafür zu sein scheint, ob mehrfach Lösungen einreicht werden oder nicht. Als Erstes wird hierzu die Multiple-Choice-Aufgabe (MC) betrachtet. Diese ist ein geschlossener Aufgabentyp, d.h., man kann in endlich vielen Versuchen die richtige Lösung finden. Ausschlaggebend dabei ist noch, dass man so lange ausprobieren kann, ohne auch nur eine Ahnung von der richtigen Antwort zu haben und trotzdem ist die Aufgabe am Ende gelöst. Diese wurde von 75% aller StudienanfängerInnen versucht richtig zu lösen. Wenn man sich das Endergebnis noch genauer anschaut, lösten alle bis auf eine Person diese Aufgabe richtig. Da in diesem Themengebiet nur eine Multiple-Choice-Aufgabe vorkommt, kann dieses Ergebnis jedoch unter keinen Umständen als signifikant betrachtet werden.

Trotzdem lohnt sich ein Vergleich mit den anderen Aufgabentypen. Deutlich anders sieht es nämlich bei dem Aufgabentyp Fill-In aus. Hier muss zum einen ohne Vorgaben der richtige Lösungsweg gefunden werden und zum anderen muss die Lösung richtig eingetragen werden. Zum Teil meinten die StudienanfängerInnen zwar das Richtige, trugen aber ihre Antwort nicht mathematisch korrekt ein. Die meisten dieser Fehler sind fundamentaler Art und nicht von der Eingabesyntax abhängig. Aufgrund des mangelnden Verständnisses für die Bedeutung mathematisch korrekter Eingaben kam bei den StudienanfängerInnen zum Teil eine hohe Frustration auf und es wurde nicht weiter versucht, die richtige Lösung zu finden. Insgesamt versuchten dies 30,83% der StudienanfängerInnen bei Fill-In-Aufgaben, nachdem sie zuvor eine fehlerhafte Lösung eingereicht hatten. Bei den mehrstufigen Aufgaben ist diese Zahl etwas größer. Hier versuchten 36,32% der StudienanfängerInnen, die richtige Lösung zu finden. Diese etwas bessere Zahl könnte man darauf zurückführen, dass hier der „Skip“-Button den

6 Erfahrungen aus dem Vorkurs

StudienanfängerInnen hilft, wenn sie einen Teil der Aufgaben nicht lösen können. Dadurch könnten sie motivierter an die Aufgaben herangegangen sein und stärker die richtige Lösung gesucht haben. Unter den hier betrachteten mehrstufigen Aufgaben war nur eine, die auch eine Multiple-Choice-Frage enthielt, d.h., alle anderen waren ausschließlich aus Fill-In-Aufgaben aufgebaut, so dass die Ergebnisse tatsächlich mit denen einzelner Fill-In-Aufgaben verglichen werden können. In der Regel haben die StudienanfängerInnen, die sich bemüht haben die richtige Lösung zu finden, den „Skip“-Button benutzt, falls sie nicht erfolgreich waren.

7 Fazit und Ausblick

Aus den gewonnenen Erfahrungen im Vorkurs sind neue Ideen und Maßnahmen für das zweite Projektjahr entstanden, wie bspw. das ohne einen Formeleditor Schwierigkeiten und Missverständnisse bei der Eingabe der Lösungen durch die Studierenden besteht. Auch einen Aufgabeneditor soll es in Zukunft geben.

7.1 Bilanz des ersten Projektjahres

Das erste Projektjahr ist insgesamt betrachtet erfolgreich gewesen. Die genauen Maßnahmenplanungen und Überlegungen wurden Schritt für Schritt realisiert. Ein grundlegendes System, wie es angedacht war, wurde erfolgreich umgesetzt und ist im laufenden Betrieb. Ein Mathematikvorkurs mit Präsenzveranstaltung und Lerneinheiten im System wurde angeboten.

7.2 Anstehende Arbeiten im zweiten Projektjahr

Sowohl aus dem allgemeinen Projektplan als auch aus den Erfahrungen aus dem ersten Projektjahr ergeben sich eine Reihe von Arbeiten, die im zweiten Projektjahr durchgeführt werden sollen und die sich auf alle drei Maßnahmenpakete verteilen.

7.2.1 Kommende inhaltliche Arbeiten

Auf der inhaltlichen Ebene sieht der Projektplan vor, im zweiten Projektjahr weitere Aufgaben zu entwerfen und vorhandene Aufgaben zu pflegen und zu verbessern (Maßnahme A III). Zudem soll im zweiten Projektjahr ein Tracking-Konzept entworfen werden (Maßnahme A IV), mit dem eine Auswertung und Steuerung des Lernprozesses der Studierenden möglich ist. Aus den Erfahrungen im ersten Projektjahr und insbesondere bei der Erstellung der Aufgaben ergibt sich, dass auf inhaltlicher Ebene zudem „Best Practices“ zur Erstellung von Aufgaben entwickelt werden können, die bei der weiteren Qualitätssicherung der vorhandenen Aufgaben sowie bei der Erstellung neuer Inhalte als Referenz dienen können.

7.2.2 Kommende technische Arbeiten

Auf technischer Ebene stehen im zweiten Projektjahr die Maßnahmen B II, B III und B V im Vordergrund, d.h. insbesondere die Implementierung weiterer Prüfkomponten und die Entwicklung eines Frameworks für geführtes Lernen, welches den Leistungsstand der Studierenden messen kann.

Unabhängig vom Projektplan wurde zudem der Bedarf für die Verbesserung der Usability von JACK identifiziert, wobei sowohl die Sicht der Studierenden (z.B. durch verbesserte Darstellung der Aufgaben und einfachere Eingabemöglichkeiten für mathematische Formeln) als auch die Sicht der Lehrenden (z.B. durch externe Editoren für Aufgaben) betroffen sind. Ein spezieller Aufgabeneditor für Lehrende könnte zudem die Anwendung der oben erwähnten „Best Practices“ unterstützen, indem er den Lehrenden entsprechende Hilfestellung und Tipps bei der Erstellung und Pflege von Aufgaben gibt.

7.2.3 Kommende organisatorische Arbeiten

Die organisatorischen Arbeiten stehen im Prinzip mit dem zweiten Projektjahr noch ganz an ihrem Anfang, können nun aber in vollem Umfang begonnen werden. Insbesondere ist vorgesehen, die Tutoren als Teil der Schulung auch aktiv in die Erstellung und Pflege von Inhalten einzubinden.

A Überblick über existierende E-Learning-Systeme

In diesem Anhang wird ein Überblick über bereits existierende E-Learning-Systeme gegeben, die durch die Recherchen der Autoren/innen gefunden worden sind.

Plattformen	Active Math	Mathe Online	Mathe Prisma	Mathe Vital
Feedback	ja	korrigiert ohne Feedback	korrigiert ohne Feedback	ohne Feedback
Aufgabentypen	Multiple-Choice, Fill-In, Essay, Zuordnung, dynamische Elemente (Applets), Concept-Map (Drag & Drop)	Multiple-Choice, Puzzle, Zuordnung, Fill-In	Multiple-Choice, Fill-In, Animation, Puzzle	Hot-Spot (Geometrie)
Anbindung an Mathetools	Jess, Problem- lösungs-, Mbase systeme	keine bekannt	keine bekannt	Cinderella
Darstellung & Eingabe	Latex, Editor	Latex	Dezimal- schreibweise	interaktive Experimente
Lernfortschritt	ja	nein	nein	nein
Zugang	Web	Web	Web	Web
Zielgruppe	Schüler, Studierende	Schüler, Studierende	Schüler, Studierende	Studierende

Tabelle A.1: Überblick über Mathematik Systeme

Um den Überblick besser zu behalten, wurden die Systeme in drei Tabellen eingeordnet. In Tabelle A.1 werden die reinen Mathematik E-Learning-Systeme vorgestellt. Diese Systeme beinhalten nur Aufgaben, die das Fach Mathematik allgemein betreffen. Dort findet man das System Active Math¹ mit dem man nicht nur vorhandene Aufgaben bearbeiten kann, sondern Lehrende auch die Möglichkeit haben Aufgaben hochzuladen und Kurse zu erstellen. Die weiteren Systeme in dieser Tabelle Mathe Online², Mathe Prisma³ und Mathe Vital⁴ haben

¹<http://www.activemath.org>

²<http://www.mathe-online.at>

³<http://www.matheprisma.uni-wuppertal.de>

⁴<http://www.mathe-vital.de>

A Überblick über existierende E-Learning-Systeme

frei verfügbare Aufgaben, die jeder bearbeiten kann, ohne sich anzumelden. Die nächste Tabelle A.2 stellt allgemeine E-Learning-Systeme dar. Diese Systeme sind themenspezifisch, d.h. sie thematisieren nur ein Themengebiet aus der Mathematik. Das System EMILeA⁵ beinhaltet reine Statistik Aufgaben und Sail-m⁶ beschäftigt sich mit mathematischen Beweisen. Weiter findet man dort noch Systeme, die die Kryptographie (Kryptol⁷) und die numerische Mathematik (NUMAS⁸) behandeln. In der letzten Tabelle A.3 werden weitere E-Learning-Systeme vorgestellt, die sich nicht auf reine Mathematik Aufgaben spezialisiert haben. Die beiden E-Learning-Systeme LON-CAPA⁹ und MUMIE¹⁰ sind ähnlich wie Active Math einzustufen nur mit dem Unterschied, dass diese sich nicht auf das Fach Mathematik beschränken. LPLUS¹¹ dagegen ist ein Prüfungssystem, mit dem man Klausuren in allen Fächern stellen kann.

Plattformen	EMILeA	Kryptol	NUMAS	Sail-m
Themen	Statistik	Kryptographie	numerische Mathematik	vollständige Induktion
Feedback	Hinweise	keins	Hinweise	Hinweise, Feedback
Aufgabentypen	Simulationen	Text Eingabe	Multiple-Choice	Beweise
Anbindung an Mathetools	nein	keine bekannt	keine bekannt	Cinderella
Darstellung & Eingabe	Editor	kein	kein	Editor
Lernfortschritt	nicht bekannt	nein	ja	nein
Zugang	Web	Offline	Web	Offline
Zielgruppe	Schüler, Studierende	Schüler, Studierende	Studierende	Studierende

Tabelle A.2: Überblick über themenspezifische E-Learning-Systeme

Zusätzlich gibt es viele Learning Management Systeme (LMS) mit denen man nicht nur Aufgaben erstellen kann, sondern auch ganze Kurse verwalten und managen. Die Lehrenden haben die Möglichkeit Kursunterlagen hochzuladen. Außerdem gibt es die Möglichkeiten von Foren und Chats mit denen Lehrende und Lernende miteinander kommunizieren können und so die Möglichkeit haben ständig in Kontakt zu stehen, aber auch Lernende untereinander. Die verschiedenen LMS unterscheiden sich unter anderem dadurch, dass einige kos-

⁵<http://www.emilea.de>

⁶<http://www.sail-m.de>

⁷<http://www.cryptool.org>

⁸<http://www.numas.de>

⁹<http://www.lon-capa.org>

¹⁰<http://www.mumie.net>

¹¹<http://www.lplus.de>

Plattformen	LON-CAPA	LPLUS	MUMIE
Aufgabenmanagement	Aufgaben in Lektionen zusammen stellen	Aufgaben selbst erstellen	Aufgaben in Lektionen zusammen stellen
Feedback	ja	ja	nicht bekannt
Aufgabentypen	Multiple-Choice, Fill-In, Essay, Zuordnung, Hot Spot, Rangordnung	Multiple-Choice, Zuordnung, Fill-In	Multiple-Choice, Fill-In, Zuordnung
Anbindung an Mathetools	MAXIMA, CAS, R Project	keine	keine
Darstellung & Eingabe	Latex, Editor	Editor	keine besondere
Lernfortschritt	ja	ja	ja
Zugang	Web	Web/Offline	Web
Zielgruppe	Schüler/Studierende aller Fächer	Prüfungen in allen Fächern	Studierende aller Fächer

Tabelle A.3: Überblick über sonstige E-Learning-Systeme

tenpflichtig sind wie z.B. Blackboard¹² und andere Open Source Produkte wie z.B. moodle¹³, ILIAS¹⁴, Stud.IP¹⁵, OLAT¹⁶, um nur einige zu nennen. Einige Learning Management Systeme bieten zusätzlich die Möglichkeit Test zur Prüfungsvorbereitung oder zur Lernerfolgskontrolle durchzuführen, wie z.B. ILIAS und moodle.

Wie man an diese Überblick erkennen kann, findet man schnell E-Learning-System, die Multiple-Choice- und Fill-In-Aufgaben beinhalten. Wenn man nun aber weiter gehen möchte und das Angebot auf mehrere Aufgabentypen und verschiedene Fächer ausweiten möchte stößt man schnell an Grenzen. Daher haben wir uns dafür entschieden das Programm JACK weiter zu programmieren. Dadurch haben wir zusätzlich die Freiheit unsere Wünsche in die Tat um zu setzen, wir können z.B. bei den Multiple-Choice-Aufgaben zu jeder Antwort-Kombination ein eigenes Feedback geben.

¹²<http://www.blackboard.com>

¹³<http://www.moodle.de>

¹⁴<http://www.ilias.de>

¹⁵<http://www.studip.de>

¹⁶<http://www.olat.de>

2012

No 53 (December 2012)

Frank, Ulrich: »Thoughts on Classification / Instantiation and Generalisation / Specialisation«

No 52 (July 2012)

Berntsson-Svensson, Richard; Berry, Daniel; Daneva, Maya; Dörr, Jörg; Fricker, Samuel A; Herrmann, Andrea; Herzwurm, Georg; Kauppinen, Marjo; Madhavji, Nazim H; Mahaux, Martin; Paech, Barbara; Penzenstadler, Birgit; Pietsch, Wolfram; Salinesi, Camille; Schneider, Kurt; Seyff, Norbert; van de Weerd, Inge (Eds.): »18th International Working Conference on Requirements Engineering – Foundation for Software Quality. Proceedings of the Workshops RE4SuSy, REEW, CreaRE, RePriCo, IWSPM and the Conference Related Empirical Study, Empirical Fair and Doctoral Symposium«

No 51 (May 2012)

Frank, Ulrich: »Specialisation in Business Process Modelling – Motivation, Approaches and Limitations«

No 50 (March 2012)

Adelsberger, Heimo; Drechsler, Andreas; Herzig, Eric; Michaelis, Alexander; Schulz, Philipp ; Schütz, Stefan; Ulrich, Udo: »Qualitative und quantitative Analyse von SOA-Studien – Eine Metastudie zu serviceorientierten Architekturen«

2011

No 49 (December 2011)

Frank, Ulrich: »MEMO Organisation Modelling Language (2) – Focus on Business Processes«

No 48 (December 2011)

Frank, Ulrich: »MEMO Organisation Modelling Language (1) – Focus on Organisational Structure«

No 47 (December 2011)

Frank, Ulrich: »Multiperspective Enterprise Modelling – Requirements and Core Diagram Typs«

No 46 (December 2011)

Frank, Ulrich: »Multiperspective Enterprise Modelling – Background and Terminological Foundation«

No 45 (November 2011)

Frank, Ulrich; Strecker, Stefan; Heise, David; Kattenstroth, Heiko; Schauer, Carola: »Leitfaden zur Erstellung wissenschaftlicher Arbeiten in der Wirtschaftsinformatik«

No 44 (September 2011)

Berenbach, Brian; Daneva, Maya; Dörr, Jörg; Fricker, Samuel; Gervasi, Vincenzo; Glinz, Martin; Herrmann, Andrea; Krams, Benedikt; Madhavji, Nazim H; Paech, Barbara; Schockert, Sixten; Seyff, Norbert (Eds.): »17th International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ 2011) – Proceedings of the REFSQ 2011 Workshops REEW, EPICAL and RePriCo, the REFSQ 2011 Empirical Track (Empirical Live Experiment and Empirical Research Fair), and the REFSQ 2011 Doctoral Symposium«

No 43 (February 2011)

Frank, Ulrich: »The MEMO Meta Modelling Language (MML) and Language Architecture. 2nd Edition«

2010

No 42 (December 2010)

Frank, Ulrich: »Outline of a Method for Designing Domain-Specific Modelling Languages«

No 41 (December 2010)

Adelsberger, Heimo; Drechsler, Andreas (Hrsg.): »Ausgewählte Aspekte des Cloud-Computing aus einer IT-Management-Perspektive – Cloud Governance, Cloud Security und Einsatz von Cloud Computing in jungen Unternehmen«

No 40 (October 2010)

Bürsner, Simone; Dörr, Jörg; Gehlert, Andreas; Herrmann, Andrea; Herzwurm, Georg; Janzen, Dirk; Merten, Thorsten; Pietschm, Wolfram; Schmid, Klaus; Schneider, Kurt; Thurimella, Anil Kumar (Eds.): »16th International Working Conference on Requirements Engineering: Foundation for Software Quality – Proceedings of the Workshops CreaRE, PLREQ, RePriCo and RESC«

No 39 (May 2010)

Strecker, Stefan; Heise, David; Frank, Ulrich: »Entwurf einer Mentoring-Konzeption für den Studiengang M.Sc. Wirtschaftsinformatik an der Fakultät für Wirtschaftswissenschaften der Universität Duisburg-Essen«

No 38 (February 2010)

Schauer, Carola: »Wie praxisorientiert ist die Wirtschaftsinformatik? Einschätzungen von CIOs und WI-Professoren«

No 37 (January 2010)

Benavides, David; Batory, Don; Grunbacher, Paul (Eds.): »Fourth International Workshop on Variability Modelling of Software-intensive Systems«

Previously published ICB Research Reports

2009

No 36 (December 2009)

Strecker, Stefan: »Ein Kommentar zur Diskussion um Begriff und Verständnis der IT-Governance – Anregungen zu einer kritischen Reflexion«

No 35 (August 2009)

Rüngeler, Irene; Tüxen, Michael; Rathgeb, Erwin P.: »Considerations on Handling Link Errors in SCTP«

No 34 (June 2009)

Karastoyanova, Dimka; Kazhamiakan, Raman; Metzger, Andreas; Pistore, Marco (Eds.): »Workshop on Service Monitoring, Adaptation and Beyond«

No 33 (May 2009)

Adelsberger, Heimo; Drechsler, Andreas; Bruckmann, Tobias; Kalvelage, Peter; Kinne, Sophia; Pellingner, Jan; Rosenberger, Marcel; Trepper, Tobias: »Einsatz von Social Software in Unternehmen – Studie über Umfang und Zweck der Nutzung«

No 32 (April 2009)

Barth, Manfred; Gadatsch, Andreas; Kutz, Martin; Ruding, Otto; Schauer, Hanno; Strecker, Stefan: »Leitbild IT-Controller/-in . Beitrag der Fachgruppe IT-Controlling der Gesellschaft für Informatik e. V.«

No 31 (April 2009)

Frank, Ulrich; Strecker, Stefan: »Beyond ERP Systems: An Outline of Self-Referential Enterprise Systems – Requirements, Conceptual Foundation and Design Options«

No 30 (February 2009)

Schauer, Hanno; Wolff, Frank: »Kriterien guter Wissensarbeit – Ein Vorschlag aus dem Blickwinkel der Wissenschaftstheorie (Langfassung)«

No 29 (January 2009)

Benavides, David; Metzger, Andreas; Eisenecker, Ulrich (Eds.): »Third International Workshop on Variability Modelling of Software-intensive Systems«

2008

No 28 (December 2008)

Goedicke, Michael; Striewe, Michael; Balz, Moritz: »Computer Aided Assessments and Programming Exercises with JACK«

No 27 (December 2008)

Schauer, Carola: »Größe und Ausrichtung der Disziplin Wirtschaftsinformatik an Universitäten im deutschsprachigen Raum – Aktueller Status und Entwicklung seit 1992«

No 26 (September 2008)

Milen, Tilev; Muller-Clostermann, Bruno: » CapSys: A Tool for Macroscopic Capacity Planning«

No 25 (August 2008)

Eicker, Stefan; Spies, Thorsten; Tschersich, Markus: »Einsatz von Multi-Touch beim Soft-waredesign am Beispiel der CRC Card-Methode«

No 24 (August 2008)

Frank, Ulrich: »The MEMO Meta Modelling Language (MML) and Language Architec-ture – Revised Version«

No 23 (January 2008)

Sprenger, Jonas; Jung, Jürgen: »Enterprise Modelling in the Context of Manufacturing – Outline of an Approach Supporting Production Planning«

No 22 (January 2008)

Heymans, Patrick; Kang, Kyo-Chul; Metzger, Andreas, Pohl, Klaus (Eds.): »Second In-ternational Workshop on Variability Modelling of Software-intensive Systems«

2007

No 21 (September 2007)

Eicker, Stefan; Nagel, Annett; Schuler, Peter M.: »Flexibilität im Geschäftsprozess-management-Kreislauf«

No 20 (August 2007)

Blau, Holger; Eicker, Stefan; Spies, Thorsten: »Reifegradüberwachung von Software«

No 19 (June 2007)

Schauer, Carola: »Relevance and Success of IS Teaching and Research: An Analysis of the Relevance Debate«

No 18 (May 2007)

Schauer, Carola: »Rekonstruktion der historischen Entwicklung der Wirtschaftsinforma-tik: Schritte der Institutionalisierung, Diskussion zum Status, Rahmenempfehlungen für die Lehre«

No 17 (May 2007)

Schauer, Carola; Schmeing, Tobias: »Development of IS Teaching in North-America: An Analysis of Model Curricula«

Previously published ICB Research Reports

No 16 (May 2007)

Müller-Clostermann, Bruno; Tilev, Milen: »Using G/G/m-Models for Multi-Server and Mainframe Capacity Planning«

No 15 (April 2007)

Heise, David; Schauer, Carola; Strecker, Stefan: »Informationsquellen für IT-Professionals – Analyse und Bewertung der Fachpresse aus Sicht der Wirtschaftsinformatik«

No 14 (March 2007)

Eicker, Stefan; Hegmanns, Christian; Malich, Stefan: »Auswahl von Bewertungsmethoden für Softwarearchitekturen«

No 13 (February 2007)

Eicker, Stefan; Spies, Thorsten; Kahl, Christian: »Softwarevisualisierung im Kontext serviceorientierter Architekturen«

No 12 (February 2007)

Brenner, Freimut: »Cumulative Measures of Absorbing Joint Markov Chains and an Application to Markovian Process Algebras«

No 11 (February 2007)

Kirchner, Lutz: »Entwurf einer Modellierungssprache zur Unterstützung der Aufgaben des IT Managements - Grundlagen, Anforderungen und Metamodell«

No 10 (February 2007)

Schauer, Carola; Strecker, Stefan: »Vergleichende Literaturstudie aktueller einführender Lehrbücher der Wirtschaftsinformatik: Bezugsrahmen und Auswertung«

No 9 (February 2007)

Strecker, Stefan; Kuckertz, Andreas; Pawlowski, Jan M.: »überlegungen zur Qualifizierung des wissenschaftlichen Nachwuchses: Ein Diskussionsbeitrag zur (kumulativen) Habilitation«

No 8 (February 2007)

Frank, Ulrich; Strecker, Stefan; Koch, Stefan: »Open Model - Ein Vorschlag für ein Forschungsprogramm der Wirtschaftsinformatik (Langfassung)«

2006

No 7 (December 2006)

Frank, Ulrich: »Towards a Pluralistic Conception of Research Methods in Information Systems Research«

No 6 (April 2006)

Frank, Ulrich: »Evaluation von Forschung und Lehre an Universitäten – Ein Diskussionsbeitrag«

No 5 (April 2006)

Jung, Jürgen: »Supply Chains in the Context of Resource Modelling«

No 4 (February 2006)

Lange, Carola: »Development and status of the Information Systems / Wirtschaftsinformatik discipline: An interpretive evaluation of interviews with renowned researchers, Part III - Results Wirtschaftsinformatik Discipline«

2005

No 3 (December 2005)

Lange, Carola: »Development and status of the Information Systems / Wirtschaftsinformatik discipline: An interpretive evaluation of interviews with renowned researchers, Part II - Results Information Systems Discipline«

No 2 (December 2005)

Lange, Carola: »Development and status of the Information Systems / Wirtschaftsinformatik discipline: An interpretive evaluation of interviews with renowned researchers, Part I - Research Objectives and Method«

No 1 (August 2005)

Lange, Carola: »Ein Bezugsrahmen zur Beschreibung von Forschungsgegenständen und -methoden in Wirtschaftsinformatik und Information Systems«

Research Group	Core Research Topics
Prof. Dr. H. H. Adelsberger Information Systems for Production and Operations Management	E-Learning, Knowledge Management, Skill-Management, Simulation, Artificial Intelligence
Prof. Dr. F. Ahlemann Information Systems and Strategic Management	Strategic planning of IS, Enterprise Architecture Management, IT Vendor Management, Project Portfolio Management, IT Governance, Strategic IT Benchmarking
Prof. Dr. P. Chamoni MIS and Management Science / Operations Research	Information Systems and Operations Research, Business Intelligence, Data Warehousing
Prof. Dr. K. Echtle Dependability of Computing Systems	Dependability of Computing Systems
Prof. Dr. S. Eicker Information Systems and Software Engineering	Process Models, Software-Architectures
Prof. Dr. U. Frank Information Systems and Enterprise Modelling	Enterprise Modelling, Enterprise Application Integration, IT Management, Knowledge Management
Prof. Dr. M. Goedicke Specification of Software Systems	Distributed Systems, Software Components, CSCW
Prof. Dr. V. Gruhn Software Engineering	Design of Software Processes, Software Architecture, Usability, Mobile Applications, Component-based and Generative Software Development
PD Dr. C. Klüver Computer Based Analysis of Social Complexity	Soft Computing, Modeling of Social, Cognitive, and Economic Processes, Development of Algorithms
Prof. Dr. T. Kollmann E-Business and E-Entrepreneurship	E-Business and Information Management, E-Entrepreneurship/E-Venture, Virtual Marketplaces and Mobile Commerce, Online-Marketing
Prof. Dr. K. Pohl Software Systems Engineering	Requirements Engineering, Software Quality Assurance, Software-Architectures, Evaluation of COTS/Open Source-Components
Prof. Dr. R. Unland Data Management Systems and Knowledge Representation	Data Management, Artificial Intelligence, Software Engineering, Internet Based Teaching
Prof. Dr. S. Zelewski Institute of Production and Industrial Information Management	Industrial Business Processes, Innovation Management, Information Management, Economic Analyses