

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

Research Report

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

ICB-Research Report, No. 59

Provided in Cooperation with:

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

Suggested Citation: Goedicke, Michael; Kurz-Karaoglu, Filiz; Schwinning, Nils; Schypula, Melanie; Striewe, Michael (2014) : Zweiter Jahresbericht zum Projekt "Bildungsgerechtigkeit im Fokus" (Teilprojekt 1.2 - "Blended Learning") an der Fakultät für Wirtschaftswissenschaften, ICB-Research Report, No. 59, Universität Duisburg-Essen, Institut für Informatik und Wirtschaftsinformatik (ICB), Essen

This Version is available at:

<https://hdl.handle.net/10419/99498>

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

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



Zweiter Jahresbericht zum Projekt „Bildungsgerechtigkeit im
Fokus“ (Teilprojekt 1.2 - „Blended Learning“) an der Fakultät 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:

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

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

goedicke@s3.uni-due.de
filiz.kurt-karaoglu@s3.uni-due.de
nils.schwinning@s3.uni-due.de
melanie.schypula@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 berichtet über technische und inhaltliche Entwicklungen sowie Einsatzerfahrungen aus dem zweiten Projektjahr des Projekts „Bildungsgerechtigkeit im Fokus, Teilprojekt 1.2 - Blended Learning“. Der Report gibt einen Überblick über die Erweiterungen, die am E-Assessment-System JACK vorgenommen wurden und beschreibt praktische Erfahrungen, die aus dem Einsatz des Systems in drei Lehrveranstaltungen und einem Mathematik-Vorkurs gewonnen werden konnten.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Stand des Projekts	1
1.2	Ziele des zweiten Projektjahres	2
1.3	Aufbau des Technical Report	4
2	Aufgabenformate und Fragentypen	5
2.1	Formularbasiertes, mehrstufiges Aufgabenformat	5
2.2	Fragentypen mit Graphen	8
2.3	Fragentypen mit Tabellen	13
3	Technische Weiterentwicklung	19
3.1	Formeleditor	19
3.2	Anbindung an Computer-Algebra-Systeme	21
3.3	OPENMATH	22
3.4	MOODLE-Integration	24
4	Praktischer Einsatz	26
4.1	Erfahrungen aus dem vorlesungsbegleitenden Einsatz	26
4.1.1	Sommersemester 2013	26
4.1.2	Wintersemester 2013/14	27
4.2	Erfahrungen aus den Vorkursen	28
4.2.1	Veröffentlichung zum Vorkurs 2012	28
4.2.2	Vorkurs 2013	29
5	Fazit und Ausblick	33
5.1	Bilanz des zweiten Projektjahres	33
5.2	Anstehende Arbeiten im drittem Projektjahr	33
5.2.1	Kommende inhaltliche Arbeiten	34
5.2.2	Kommende technische Arbeiten	34
5.2.3	Kommende organisatorische Arbeiten	35
A	Aufgabenbeispiele	36
	Literaturverzeichnis	46

Abbildungsverzeichnis

2.1	Lösungsweg Lineares Gleichungssystem	6
2.2	Der Graph einer linearen Aufgabe	7
2.3	Der Graph einer Aufgabe mit mehreren Pfaden	8
2.4	Der Graph einer Aufgabe mit Zyklus	9
2.5	Aufgabe mit Graphen im Feedback	10
2.6	Aufgabe mit Graphen als Antwortoption	11
2.7	Beispiel für den Java Script Code, um einen Graphen darstellen zu können. . .	12
2.8	Graph zum Java Script Code	12
2.9	Eine Tabellen Aufgabe zum anklicken	14
2.10	Beispiel für den XML-Code, um eine SCList Aufgabe darstellen zu können. . .	15
2.11	Eine Tabellen Aufgabe mit Lückentext	17
2.12	Beispiel für den XML-Code für die Regeln.	17
2.13	Beispiel für den XML-Code für die Regelverletzung.	18
3.1	Der Formeleditor in Benutzung	20
3.2	Beispiel für einen Ausdruck in OPENMATH	23
3.3	Architektur von JACK bezüglich der Handhabung mathematischer Eingaben .	23
3.4	Bewertungsübersicht für die Lehrenden in MOODLE 2	25
3.5	Bewertungsübersicht für die Lernenden in MOODLE 2	25
4.1	Graphische Darstellung der Anzahl der Bearbeitungen	31

1 Einleitung

Der vorliegende Bericht befasst sich mit den Fortschritten des zweiten Projektjahres des Projekts „Bildungsgerechtigkeit im Fokus, Teilprojekt 1.2 - Blended Learning“, das im Rahmen des Bund-Länder-Programms „Qualität der Lehre“ gefördert wird. Ziel des Projektes ist es, insbesondere durch Verbesserungen in der Studieneingangsphase die Studienbedingungen und die Qualität der Lehre zu verbessern. An der Fakultät Wirtschaftswissenschaften hat besonders das Fach Mathematik die Studierenden vor einer besonderen Herausforderung gestellt, weswegen dieses Teilprojekt genau dort angesiedelt ist. Insbesondere wird dabei die Mathematik-Grundausbildung fokussiert.

Der vorliegende Bericht schließt an den Anfang 2013 erschienenen ersten Jahresbericht [FGKK⁺13] an und thematisiert insbesondere die technischen Erweiterungen am E-Learning-System JACK sowie die weiteren Nutzungserfahrungen aus dem praktischen Einsatz in den Lehrveranstaltungen der Fakultät.

1.1 Stand des Projekts

Im ersten Projektjahr wurde das bereits für die Programmierungsvorlesung erfolgreich eingesetzte automatisierte Lehr- und Bewertungssystem JACK dahin gehend weiterentwickelt, dass es auch Mathematikaufgaben anbieten kann. Dazu wurden zunächst einfache Fragentypen wie Multiple-Choice und Fill-In technisch umgesetzt und das System um mathematikspezifische Eigenschaften wie die Anzeige von Formeln durch $\LaTeX$ sowie die Erstellung von Aufgaben mit zufälligen Zahlenwerten ergänzt. Ferner wurde mit einer Anbindung an das an der Universität Duisburg-Essen eingesetzte Lern-Management-System MOODLE 2 begonnen.

Neben diesen technischen Arbeiten wurde ein Aufgabenpool von ca. 200 Mathematikaufgaben erstellt, der die Kompetenzbereiche der Sekundarstufe 1 abdeckt. Diese wurden in einem Mathematik-Vorkurs zu Beginn des Wintersemesters 2012/2013 mit 150 Teilnehmern in einem Blended-Learning-Szenario eingesetzt.

1.2 Ziele des zweiten Projektjahres

Die Zielsetzungen für das zweite Projektjahr ergeben sich sowohl aus der grundsätzlichen Planung des Projektes über alle Projektjahre hinweg, als auch aus den Erfahrungen des ersten Projektjahres. Insbesondere ergeben sich aus diesen Erfahrungen sowie den Erfordernissen des Weiteren praktischen Einsatzes des Systems im zweiten Projektjahr Prioritäten, die einen Einfluss auf die Details der Arbeit im zweiten Projektjahr haben, so dass einige der im ursprünglichen Projektplan vorgesehenen Ziele noch nicht erreicht wurden.

- **Weitere Aufgaben sind entworfen:**

Da im ersten Projektjahr bereits ein umfangreicher Pool von ca. 200 Aufgaben erstellt worden war, konnte im zweiten Projektjahr die Arbeit in diesem Bereich stärker auf die Detailarbeit an den Aufgaben fokussiert werden. Dementsprechend ist durch studentische Hilfskräfte eine verhältnismäßig geringe Menge von 30 Aufgaben zusätzlich entworfen worden, die in den vorhandenen Aufgabenpool einfließen. Zusätzlich wurden jedoch alle vorhandenen Aufgaben durchgesehen und bei Bedarf auf Basis der Erfahrungen aus dem Vorkurs optimiert, um eine hohe Qualität des vorhandenen Aufgabenpools sicherzustellen.

- **Weitere Prüfkomponenten sind implementiert:**

Sowohl aus den Erfahrungen mit der Erstellung von Aufgaben aus dem Vorkurs, als auch aus den Anforderungen an einen Einsatz des Systems in regulären Lehrveranstaltungen ergaben sich schnell Wünsche nach einer Erweiterung der vorhandenen Aufgabentypen. Entsprechend dieser Wünsche wurden die Fähigkeiten der vorhandenen Prüfkomponenten im Verlauf des zweiten Projektjahres schrittweise ausgebaut, um insbesondere komplexere Zusammenhänge im Rahmen von Multiple-Choice- und Fill-In-Aufgaben abbilden zu können.

- **Eine zweite Auflage des Mathematik-Vorkurses wurde durchgeführt:**

Zu Beginn des Wintersemesters 2013/2014 wurde erneut ein Vorkurs für Mathematik an der Fakultät für Wirtschaftswissenschaften organisiert. Im Gegensatz zum Vorjahr wurde er diesmal mit ca. 200 Studierende als reine E-Learning-Veranstaltung ohne Präsenzanteil durchgeführt. Dazu wurde insbesondere ein Skript der Kursinhalte verfasst und den Studierenden zur Verfügung gestellt.

- **Der Einsatz von JACK in regulären Lehrveranstaltungen hat begonnen:**

Im Sommersemester 2013 wurde JACK in der regulären Lehrveranstaltung „Mikroökonomie I“ von Prof. Amann eingesetzt. Dazu wurden vorab insbesondere mehrere Mitarbeiterinnen und Mitarbeiter des Lehrstuhls geschult, Aufgaben und Kurse zu erstellen. Insgesamt wurden 31 verschiedene Aufgaben in verschiedenen Fragentypen

entworfen, die für vorlesungsbegleitende Übungen und in 5 Testate im Laufe des Semesters eingesetzt worden sind.

Im Wintersemester 2013/2014 wurde dieses Konzept auf die Veranstaltungen „Mikroökonomie II“ sowie die Vorlesung „Deskriptive Statistik“ von Prof. Hanck erweitert und erfolgreich umgesetzt. Dazu wurde erneut eine Schulung für Mitarbeiterinnen und Mitarbeiter der Lehrstühle durchgeführt.

- **Es existiert noch kein vollständiges Tracking-Konzept zur Beobachtung des Lernfortschritts:**

Entgegen der ursprünglichen Projektplanung konnten die Arbeiten an einem Tracking-Konzept zur Beobachtung des Lernfortschritts noch nicht abgeschlossen werden. Es ist für Lehrende bereits möglich, erreichte Punktzahlen einzusehen und dabei auch den zeitlichen Verlauf der Einreichungen von Lösungen zu berücksichtigen. Darüber hinaus wurden bisher Grundlagen erarbeitet, wie durch statistische Verfahren Schwierigkeitsparameter von Aufgaben und Leistungsparameter von Studierenden bestimmt werden können, was die Basis für eine detailliertere und aufgabenübergreifende Beobachtung des Lernfortschritts sein kann. Es ist vorgesehen, die konzeptionellen Arbeiten im dritten Projektjahr fortzusetzen und abzuschließen.

- **Das Prüfungssystem bietet noch keine automatische Führung durch den Lernstoff an:**

Die Arbeiten an einer automatischen Führung durch den Lernstoff wurden zugunsten des praktischen Einsatzes des bisherigen Systemstandes zurückgestellt. So konnten mehr Erfahrungen über das Nutzungsverhalten und die Wünsche der Studierenden gesammelt werden, die im dritten Projektjahr in die Planung der automatischen Führung durch den Lernstoff fließen sollen. Auch die oben genannten Schwierigkeitsparameter und Leistungsparameter sollen dabei berücksichtigt werden.

- **Ein Netzwerk aus geschulten Tutoren befindet sich im Aufbau:**

Durch den Mathematik-Vorkurs im Wintersemester 2012/2013 sowie die Mitarbeit in der Durchführung der regulären Lehrveranstaltungen konnten mehrere studentische Hilfskräfte Erfahrungen in der Betreuung von Blended-Learning-Szenarien sammeln, bei denen die Aufgaben über JACK abgewickelt werden. Die studentischen Hilfskräfte wurden dabei zur Erstellung und der Pflege von Aufgaben, Beantwortung von Fragen über die Foren von Lern-Management-Systemen sowie die Durchführung von Tutorien eingesetzt.

- **Wissenschaftliche Ergebnisse wurden publiziert:**

Auf der 11. e-Learning Fachtagung Informatik in Bremen wurde ein Beitrag mit dem Titel „Beobachtungen zur Motivation der Studierenden bei verschiedenen Frageformaten“ publiziert. Dort wurde anhand des Vorkurses aus dem Wintersemester 2012/2013

1 Einleitung

untersucht, ob ein Zusammenhang zwischen der Motivationen der Studierenden und den verschiedenen Frageformaten besteht.

1.3 Aufbau des Technical Report

Der vorliegende Technical Report gliedert sich wie folgt: In Kapitel 2 werden die inhaltlichen Erweiterungen von JACK in Form von Aufgabenformaten und Fragentypen diskutiert. Kapitel 3 widmet sich der technischen Weiterentwicklungen im Bezug auf die Integration von Computer-Algebra-Systemen und der Vervollständigung der MOODLE-Integration. Kapitel 4 berichtet aus dem praktischen Einsatz von JACK in den verschiedenen Lehrveranstaltungen der Fakultät. Der Technical Report schließt mit einem Fazit in Kapitel 5. Anhang A enthält als Beispiel für konkrete Inhalte in JACK eine Übersicht über die Aufgaben des Mathematikvorkurses der Fakultät für Wirtschaftswissenschaften.

2 Aufgabenformate und Fragentypen

Im zweiten Projektjahr wurde ein neuer Aufgabentyp entwickelt der „Formbased“ genannt wurde und u.a. die im ersten Jahr entwickelte Aufgabentypen „Fill-In“ und „Multiple-Choice“ beinhaltet. Zu diesem Aufgabentyp gibt es verschiedene Fragentypen, wie die zuvor genannten beiden und noch den Aufgabentyp SC-Liste, auf den später genauer eingegangen wird. Dieser neue Aufgabentyp besteht aus mehreren Stufen, die je nach Wunsch erstellt werden kann, d.h., die erste Teilaufgabe in einer Aufgabe ist z.B. eine „Multiple-Choice“ Aufgabe und die folgende Teilaufgabe eine „Fill-In“ Aufgabe.

2.1 Formularbasiertes, mehrstufiges Aufgabenformat

Während im ersten Projektjahr noch zwischen den Aufgabentypen „Multiple-Choice“, „Fill-In“ und mehrstufigen Aufgaben, bei denen man eine Sequenz von Aufgaben der zuvor genannten Typen definieren konnte, unterschieden wurde, gibt es seit dem zweiten Projektjahr nur noch einen einzigen Aufgabentyp, welcher die drei bisherigen Typen vereint. Dieser neue Aufgabentyp trägt den Namen „FormBased“. Eine Aufgabe dieses Typs besteht aus einer beliebigen Anzahl von so genannten „Stufen“. Eine jede Stufe ist eine in sich geschlossene Aufgabe, welche der Lernende lösen muss und zu welcher er vom System Feedback bekommt. Die verschiedenen Arten, wie eine Lösung zu einer Stufe einzugeben ist, werden ab jetzt als „Fragentypen“ bezeichnet. Mögliche Fragentypen sind die aus dem ersten Projektjahr bekannten Formate „Multiple-Choice“ und „Fill-In“, sowie der neu entwickelte Fragentyp „Tabelle“, welcher in Abschnitt 2.3 detailliert beschrieben wird. Eine Aufgabe vom Typ „FormBased“ lässt sich visualisieren als ein gerichteter Graph, wobei jede Stufe der Aufgabe einem Knoten des Graphen entspricht. Kanten sind zwischen je zwei Stufen der Aufgabe erlaubt. Jede Stufe besitzt sogar mindestens eine ausgehende Kante, welche den Defaultwert für die nächste Stufe oder das Ende der Aufgabe definiert. Zudem können weitere Kanten ergänzt werden, welche dann mit einer Bedingung versehen werden müssen. Die Bedingung gibt an, wann diese Kante der Defaultkante vorzuziehen ist. Eine solche Bedingung kann von den folgenden Parametern abhängen:

- **Die Eingabe des Lernenden:** Ein wichtiger Bestandteil des Aufgabentyps „FormBased“ ist, dass in der Folgestufe mit der Eingabe des Lernenden weiter gearbeitet werden kann. Ist die Aufgabenstellung zudem offen, d.h. es gibt unendlich viele richtige Lösungen,

2 Aufgabenformate und Fragentypen

$$\begin{array}{rclcrcl} 3x & + & 6y & = & 54 \\ 5x & - & 3y & = & -1 \\ \hline 13x & & & = & 52 \\ 5x & - & 3y & = & -1 \\ \hline x & & & = & 4 \\ 5x & - & 3y & = & -1 \\ \hline x & & & = & 4 \\ & - & 3y & = & -21 \\ \hline x & & & = & 4 \\ & & y & = & 7 \end{array}$$

Abbildung 2.1: Ein möglicher Lösungsweg eines Linearen Gleichungssystems. Die Aufgabe ist beendet, wenn die Variablen x und y isoliert und normiert sind und ihre korrekten Werte auf der rechten Seite der Gleichungen stehen.

kann es notwendig sein, auf die Eingabe des Lernenden angemessen reagieren zu können. Muss der Lernende zum Beispiel in Stufe 1 eine quadratische Funktion angeben, sodass eine vorgegebene Gerade die Tangente an diese in einem ebenfalls vorgegebenen Punkt ist, so kann es passieren, dass die vom Lernenden eingegebene Lösung keine, eine oder zwei Nullstellen hat. Eine mögliche nächste Aufgabe könnte also die Berechnung der Nullstellen sein, falls welche vorhanden sind, während man direkt zur Berechnung des Scheitelpunktes der quadratischen Funktion übergehen kann, falls keine Nullstellen vorhanden sind.

- **Die Punktzahl des Lernenden:** Aus didaktischer Sicht kann es sinnvoll sein, wenn das System unterschiedlich reagiert, je nachdem ob der Lernende die richtige Lösung zu einer Stufe gefunden hat oder nicht. Ein Beispiel ist das Prinzip der „Probe“, welches besonders in mündlichen Prüfungen häufig angewendet wird und vom System simuliert werden kann.
- **Die Belegung der Variablen:** In JACK können Autoren generische Aufgaben erstellen. Dies sind Aufgaben, bei denen einige Werte der Aufgabenstellung erst im Moment der Anzeige generiert werden. Ein Beispiel für eine solche Aufgabe ist die Kurvendiskussion eines Polynoms vom Grad 3, bei dem die Koeffizienten des Polynoms variabel sind.

Das System kann somit auf die Eingabe der Lernenden reagieren und somit besser auf diese eingehen. Dadurch, dass mit den Eingaben der Lernenden weiter gearbeitet werden kann, können diese bei manchen Aufgaben sogar ihr Arbeitstempo selbst bestimmen. Ein Beispiel für eine solche Aufgabe ist das Lösen eines Linearen Gleichungssystems (LGS) mit dem Gauß-Verfahren. Dieses muss mit Hilfe von vorgegebenen Operationen so lange umgeformt werden, bis die Lösungen der Variablen gefunden sind (siehe Abbildung 2.1).

Der Lösungsweg bei einer solchen Aufgabe ist jedoch nicht vorgegeben, sondern es gibt eine große Anzahl verschiedener Lösungswege. Das System prüft daher lediglich, ob der Lernende

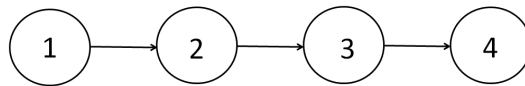


Abbildung 2.2: Der Graph einer Aufgabe mit vier Stufen. Der Verlauf des Graphen ist linear, d.h. es gibt keine Abzweigungen. Bearbeiten die Lernenden die n -te Stufe, so müssen sie anschließend die $n + 1$ -te Stufe bearbeiten.

eine gültige Umformung gemacht hat oder nicht und ob das Ende der Aufgabe erreicht ist. Eine gültige Umformung bedeutet dabei eine beliebige Anwendung von Operationen des Gauß-Verfahrens, es entsteht also ein neues Lineares Gleichungssystem, welches die gleiche Lösungsmenge besitzt. Das Ende der Aufgabe ist erreicht, wenn alle Variablen isoliert und normiert sind und ihre korrekten Werte auf der rechten Seite der Gleichungen stehen. Wurde eine gültige Umformung gemacht, das Ende der Aufgabe jedoch noch nicht erreicht, so erhalten die Lernenden die gleiche Stufe erneut, allerdings mit ihrer Eingabe aus der vorigen Stufe als zu lösendem LGS. Die Lernenden können somit selbst bestimmen, wie viele Schritte sie für die Aufgabe benötigen.

Bepunktung von mehrstufigen Aufgaben

Das Modell einer Lösung in JACK sieht es vor, dass Lernende eine Punktzahl zwischen 0 und 100 Punkten erhalten. Dieses Konzept sollte auch für mehrstufige Aufgaben beibehalten werden. Somit ist es notwendig, für die richtige Bearbeitung einer einzelnen Stufe eine Teilpunktzahl zu verteilen. Dies gestaltet sich jedoch zuweilen bei manchen Aufgaben, bei denen der Graph verschiedene Abzweigungen aufweist, als schwierig. Die Grundidee bei der Bepunktung von mehrstufigen Aufgaben ist, dass ein Lernender der alle Stufen richtig bearbeitet, die maximale Punktzahl von 100 Punkten erhält. Für jede Stufe prüft ein so genannter Checker anhand einer vom Lehrenden erstellten Vorlage, ob die eingereichte Lösung richtig oder falsch ist. Zudem verteilt der Checker für die Lösung zu einer Stufe ebenfalls eine Punktzahl zwischen 0 und 100 Punkten, wobei richtige Lösungen stets mit 100 Punkten bewertet werden. Der Checker ist sogar auch in der Lage, Teilpunkte zu verteilen. Somit erscheint es logisch, dass die Lernenden bei einem linearen Verlauf des Graphen (s. Abb. 2.2) mit n Stufen und Checkerergebnissen x_i für Stufe i die Gesamtpunktzahl

$$P = \frac{1}{n} \sum_{i=1}^n x_i$$

erhalten.

Komplizierter wird die Bepunktung einer Aufgabe, bei der es verschiedene Pfade gibt, welche zum Teil die gleichen Stufen enthalten. Ein Beispiel für eine solche Aufgabe ist in Abbildung 2.3 zu sehen. Bei dieser Aufgabe gibt es vier verschiedene Möglichkeiten, den Graphen zu

2 Aufgabenformate und Fragentypen

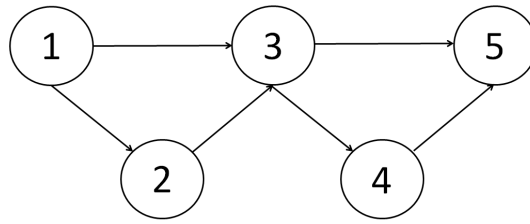


Abbildung 2.3: Der Graph einer Aufgabe mit mehreren Pfaden unterschiedlicher Länge.

durchlaufen: Dies sind die Pfade $p_1 : 1 \rightarrow 3 \rightarrow 5$, $p_2 : 1 \rightarrow 2 \rightarrow 3 \rightarrow 5$, $p_3 : 1 \rightarrow 3 \rightarrow 4 \rightarrow 5$ und $p_4 : 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$. Man beachte, dass die Pfade gleiche Teilpfade enthalten können. So enthalten bspw. p_2 und p_4 den Teilpfad $1 \rightarrow 2 \rightarrow 3$. Im Allgemeinen sollte die Bepunktung für einen zyklensfreien Pfad der Länge n und die Checkerergebnisse $x_1, \dots, x_n$ wie bei einer linearen Aufgabe $P = \frac{1}{n} \sum_{i=1}^n x_i$ betragen. Jedoch steht bei Stufen, die zu mehreren, unterschiedlich langen Pfaden gehören, bei der Bearbeitung noch nicht fest, welcher Pfad durch die Aufgabe gewählt wird und welche Teilpunktzahl für diese Stufe daher zu verteilen ist. Da den Lernenden in JACK jedoch jederzeit die aktuelle Punktzahl angezeigt werden kann, rechnet das System zunächst mit dem längstmöglichen Pfad durch die Aufgabe, ausgehend von dem bisher durchlaufenen Pfad. Wird in dem obigen Beispiel also Stufe 2 korrekt bearbeitet, so sucht das System alle Pfade, welche den Teilpfad $1 \rightarrow 2$ enthalten und verwendet den längsten dieser Pfade zur Berechnung der Teilpunktzahl. In diesem Fall sind das die Pfade p_2 und p_4 , das System verteilt für eine korrekte Bearbeitung also $\frac{100}{5} = 20$ Punkte. Dieser Vorgang wird nach jeder Einreichung für alle durchlaufenen Stufen durchgeführt. Wird anschließend doch der Pfad p_2 gewählt und die Stufe 5 bearbeitet, so ändert das System die Punktzahl für Stufe 3 auf $\frac{100}{4} = 25$ Punkte.

Enthält der Graph einen Zyklus mit den Knoten $v_1, \dots, v_n$ und $v_1 = v_n$, wie z.B. in Abb. 2.4 zu sehen, so gibt es zu jedem Pfad durch die Aufgabe, welcher den Zyklus enthält, einen Pfad durch die Aufgabe, welcher zyklensfrei ist. Man erhält diesen, indem man von jedem Zyklus die Knoten $v_1, \dots, v_{n-1}$ aus dem Pfad löscht. Da man auf beiden Pfaden bei richtiger Bearbeitung die volle Punktzahl erhalten soll, müssen die zu den Knoten $v_1, \dots, v_{n-1}$ gehörigen Stufen mit 0 Punkten bewertet werden. Das System muss also den beschrifteten Pfad analysieren und die Zyklen aus diesem herauslösen. Anschließend verteilt es für jede durchlaufene Stufe die vom Checker errechnete Punktzahl anhand der oben beschriebenen Methode.

2.2 Fragentypen mit Graphen

Ein neues Feature in JACK ist die Möglichkeit seine Aufgaben mit interaktiven Grafiken zu versehen. Im vergangenen Jahr konnte man lediglich Bilder, in denen Funktionen dargestellt

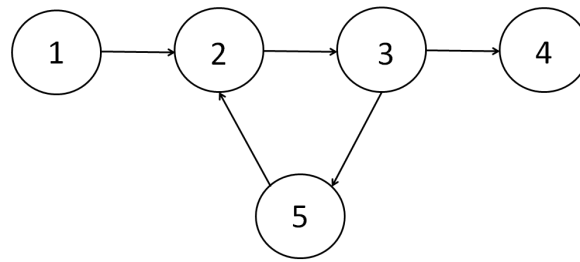


Abbildung 2.4: Der Graph einer Aufgabe mit einem Zyklus.

wurden, zu den Aufgaben hinzufügen. Diese hatten leider den Nachteil, dass sie statisch waren und dadurch weder auf Variablen eingehen konnten, noch auf die Eingabe der Studierenden. Mit den interaktiven Grafiken können nun auch die Funktionen dargestellt werden, deren Werte u.a. aus Variablen bestehen. Zudem hat man auch die Möglichkeit im Feedback die studentische Lösung zeichnen zu lassen. Dadurch können die Studierenden einfacher den Fehler ihrer angegebenen Lösung erkennen und verstehen.

In Abbildung 2.5 ist eine Aufgabe dargestellt, in der die Studierenden den Schnittpunkt von zwei Geraden bestimmen sollen. Nachdem von den Studierenden eine Eingabe getätigt wurde erscheint im Feedback eine Grafik, in der die beiden Geraden abgebildet sind. Zudem wird auch der Punkt angezeigt, der als Lösung angegeben wurde. So können die Studierenden in der Grafik ablesen, ob ihre Lösung richtig oder falsch ist. Für den Fall, dass die Lösung nicht korrekt ist können sie vor allem selber erkennen, warum die Lösung falsch ist. Dadurch stellt sich ein positiver Lerneffekt bei den Studierenden ein.

Ein weiteres Beispiel ist die Aufgabe in Abbildung 2.6. Hier ist eine Multiple-Choice Aufgabe dargestellt, die als Antwortoptionen verschiedene Grafiken zeigt. Die Frage ist welche Grafik stellt die beiden Funktionen dar. In dieser Aufgabe kann man nun ein weiteres Feature der Grafiken sehen. Die beiden Funktionen $f(x) = |x + 4|$ und $g(x) = 8$ sind mit Variablen belegt. Dies bedeutet in diesem Fall, dass die beiden Zahlen 4 in $f(x)$ und 8 in $g(x)$ sich bei jedem neuen Aufruf der Aufgabe ändern. Da die Grafiken interaktiv dargestellt werden und sie sich bei jedem neuen Aufruf der Aufgabe neu zusammenbauen, wie die Belegung der Variablen, beachten sie dies auch.

Die Grafiken werden mit JSXGraph¹ dargestellt. Man schreibt einen Java Script Code an die Stelle der Aufgabe, wo am Ende der oder die Grafiken erscheinen sollen. Die Anzahl der Grafiken in einer Aufgabe sind unbegrenzt. Zudem ist man auch nicht eingeschränkt, in welchem Teil der Aufgabe diese erscheinen sollen. Sie können im Aufgabentext, in den Antwortoptionen bei Multiple-Choice Aufgaben oder im Feedback dargestellt werden. In Abbildung 2.7 ist der Java Script Code und in Abbildung 2.8 die Grafik mit Slidern dargestellt. In der obersten Zeile gibt man der Grafik einen Namen. Dieser erscheint dann unter der Grafik.

¹<http://jsxgraph.uni-bayreuth.de/wp/>

2 Aufgabenformate und Fragentypen

Berechne den Schnittpunkt der beiden Geraden:

$$f(x) = \frac{3}{4}x + \frac{1}{4}$$

$$g(x) = -\frac{4}{5}x + 8$$

$$S = (\boxed{1} \mid \boxed{2}).$$

Punkte: 0/100



Feedback:

In dem Bild unten sind die beiden Graphen zu sehen und dein Punkt eingesetzt:

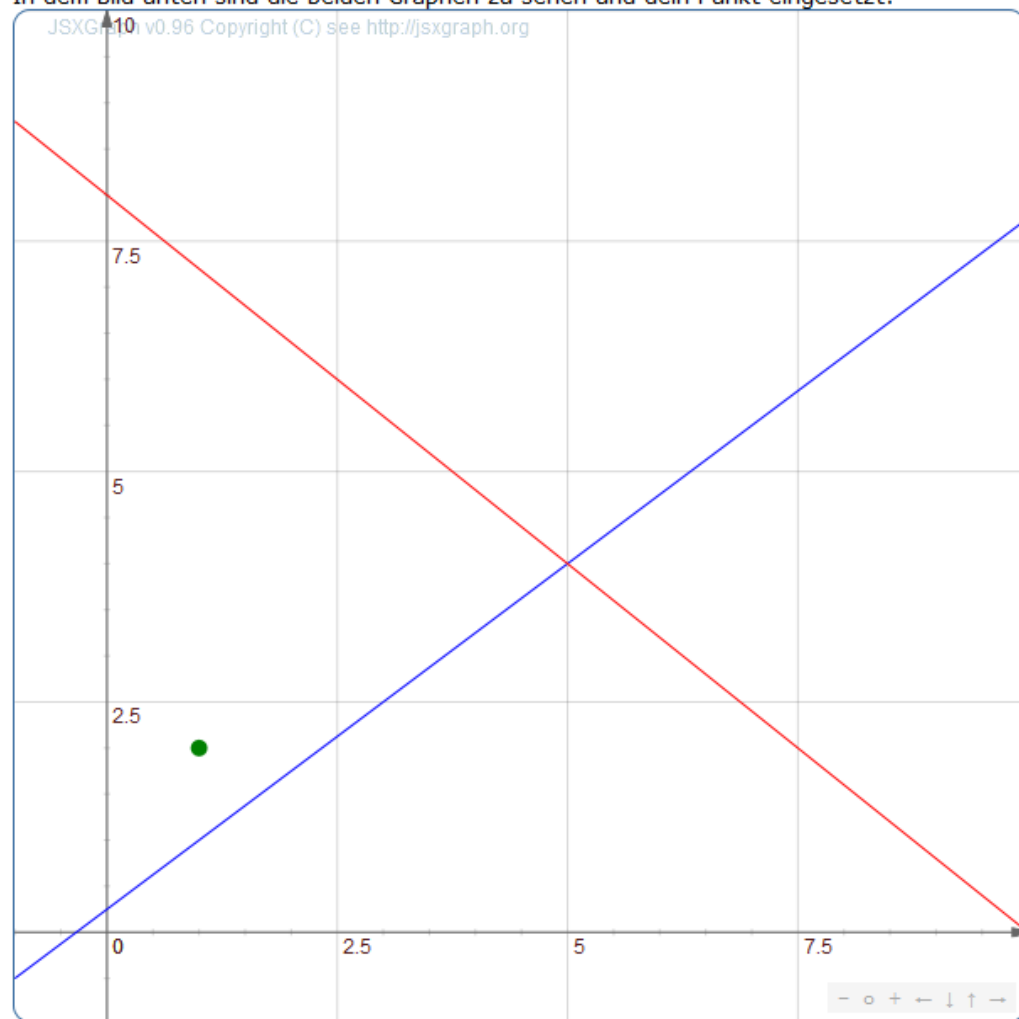


Abbildung 2.5: Screenshot einer Aufgabe, in der den Studierenden die eingegebene Lösung im Feedback in ein Koordinatensystem gezeichnet wird.

Wir wollen die beiden Seiten der Ungleichung nun als Funktionen $f(x) = |x + 4|$ (rot) und $g(x) = 8$ (grün) auffassen. Welche der folgenden Grafiken stellt dieses Szenario dar?

Antworten:

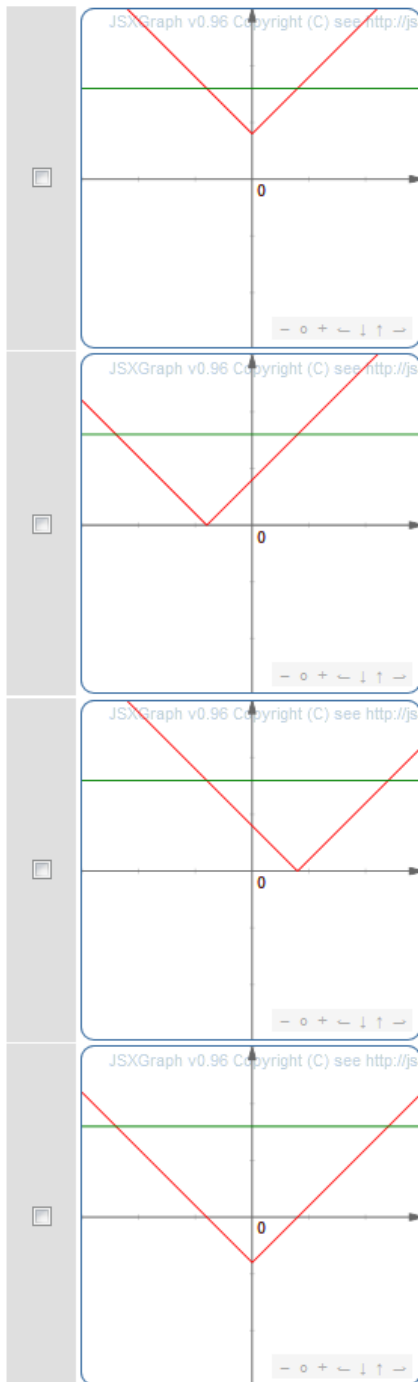


Abbildung 2.6: Screenshot einer Aufgabe, die zwei verschiedene Features von JACK nutzt: Zum einen werden verschiedene Graphen als Antwortoption in einer Multiple-Choice Aufgabe verwendet und zum anderen werden die Funktionen in den Graphen mit Variablen dargestellt.

2 Aufgabenformate und Fragentypen

```
[graph name="Parabel" id="box1" Width="400px" height="400px"]
  var board = JXG.JSXGraph.initBoard('box1', {boundingbox
    : [-10,10,10,-10], axis:true});
  var a = board.create('slider', [[0,-6],[6,-6],[-3,1,3]],name:'a');
  var b = board.create('slider', [[0,-7],[6,-7],[-3,0,3]],name:'b');
  var c = board.create('slider', [[0,-8],[6,-8],[-3,0,3]],name:'c');
  var graph = board.create('functiongraph', [function(x) return a.
    Value()*x*x + b.Value()*x + c.Value();]);
[/graph]
```

Abbildung 2.7: Beispiel für den Java Script Code, um einen Graphen darstellen zu können.

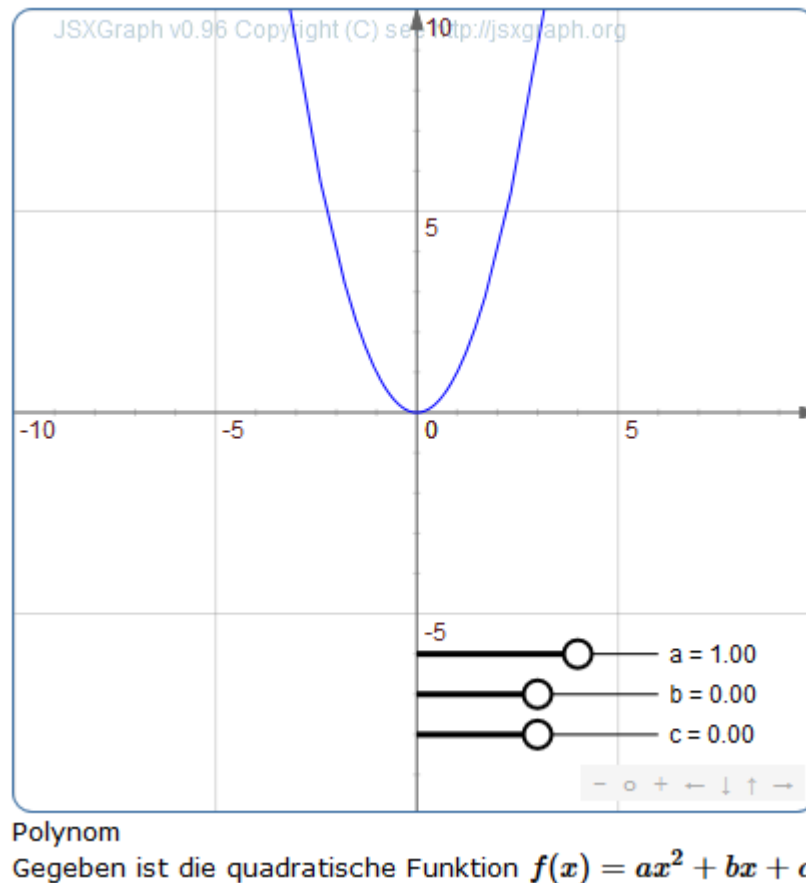


Abbildung 2.8: Der zugehörige Graph zu dem Java Script Code in Abbildung 2.7. Die Slider lassen sich nach links und rechts verschieben. Dadurch verändert sich die quadratische Funktion.

Jeder Graph einer Aufgabe braucht eine eindeutige id. Da ist besonders darauf zu achten, wenn in einer Aufgabe mehr als ein Graph angezeigt wird. Zum Schluss kann man die Größe der Box bestimmen, so dass sie zum Layout passt. Danach gibt man dem Koordinatensystem in der Box ein paar Eigenschaften. Man kann z.B. die Einheiten bestimmen und die Sichtbarkeit der Linien. Zum Schluss gibt man die Funktion an, die am Ende in der Aufgabe zu sehen sein soll. Dabei sind die Möglichkeiten fast unbegrenzt.

2.3 Fragentypen mit Tabellen

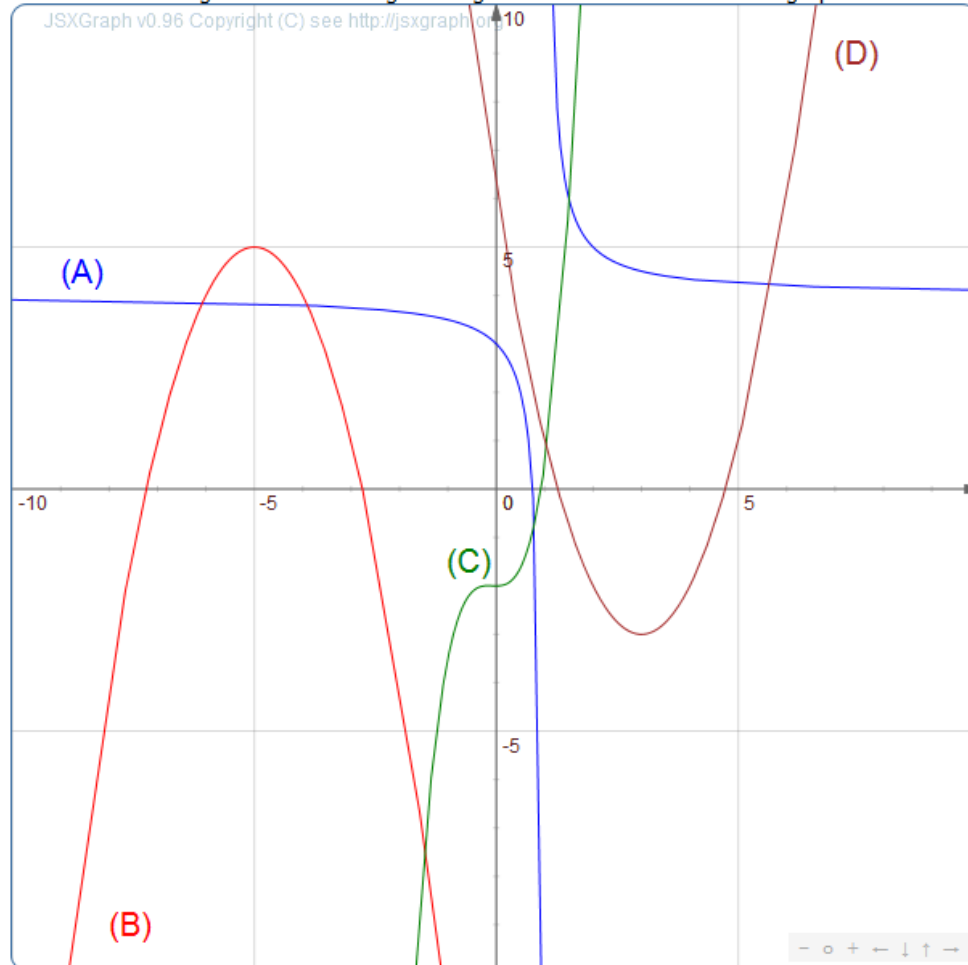
Mit der Zeit reichte der Fragentyp Multiple-Choice nicht mehr aus, es bestand u.a. das Bedürfnis nach „wahr oder falsch“-Fragen. Diese „wahr oder falsch“-Fragen wurden dann so weit ausgebaut, dass auch Aufgaben in Form einer $m \times n$ -Tabelle mit m Zeilen und n Spalten gestellt werden können. Als Beispielaufgabe dient die Aufgabe in Abbildung 2.9. In dieser Aufgabe soll jeder Funktionsgleichung der passende Graph in dem Bild zugeordnet werden. In diesem Fall ist in jeder Zeile genau ein Kästchen anzukreuzen. Eine Tabellen-Aufgabe kann auch so gestellt werden, dass in jeder Zeile unterschiedlich viele Antworten richtig sind.

In Abbildung 2.10 ist die zugehörige XML-Struktur zu der Graphen Aufgabe aus Abbildung 2.9 zu sehen. Hier sei darauf hingewiesen, dass der Code für die Darstellung des Graphen aus Platzgründen und der besseren Übersicht weggelassen wurde. Der große Unterschied zu den anderen Fragentypen findet sich in dem Tag `QUESTIONLIST`, hier wird die Tabelle beschrieben. In dem Attribut `FIRSTCOLUMN` wird die erste Spalte beschriftet, dieses kann auch leer gelassen werden, wie in diesem Beispiel. Die übrigen Spalten werden in dem Attribut `COLUMNNAMES` beschrieben, dabei ist zu beachten, dass die verschiedenen Spaltenüberschriften durch ein Komma getrennt werden. In dem Beispiel haben die Spalten die Überschriften (A), (B), (C) und (D), die für die verschiedenen Graphen stehen. Die Zeilenüberschriften werden über das Tag `QUESTION` in dem Attribut `ASSERTION` angegeben. Jede Zeile bekommt einen eigenen Tag. Wie in diesem Beispiel zu sehen ist gibt es auch bei den Zeilen und Spalten Beschriftungen die Möglichkeit $\LaTeX$ -Befehle zu benutzen. Hier wird jeweils die Funktionsgleichung angegeben, die dann dem passenden Graphen zugeordnet werden soll.

Mit den beiden Attributen `COUNT` und `FEEDBACKTYPE` werden die verschiedenen Feedback Möglichkeiten bestimmt. Je nach Größe der Tabelle wächst die Anzahl der Feedbackmöglichkeiten stark an, so dass der Arbeitsaufwand schnell groß wird manuelles Feedback zu schreiben. Die Anzahl der Möglichkeiten bei einer $m \times n$ -Tabellen Aufgabe beläuft sich auf $2^{m \cdot n}$, daher gibt es die Möglichkeit automatisches Feedback von dem System generieren zu lassen. Dabei können die Anzahl der falsch eingereichten Zeilen oder die falsch eingereichten Einträge ausgegeben werden. Bei der zeilenweisen Auswertung reicht bereits ein falscher Eintrag in der Zeile damit diese als falsch bewertet wird. Bei m Zeilen werden dann $100/m$ Punkte pro richtiger Zeile vergeben. In diesem Beispiel hier wird am Ende zurückgegeben in wie vielen Zeilen ein Fehler gemacht wurde, da bei dem `FEEDBACKTYPE` „auto“ eingetragen wurde und bei `COUNT` „row“. Des Weiteren kann der Autor einer Aufgabe zusätzlich einzelnes Feedback schreiben, wie bei den Multiple-Choice Aufgaben. Wichtig ist noch bei der Angabe der richtigen Lösung zu beachten, dass diese Angaben zeilenweise gemacht werden und durch Kommatas getrennt werden. Hier wird das selbe Muster aus Nullen und Einsen verwendet, wie bei den Multiple-Choice Aufgaben. In dem Beispiel 2.9 wird jeder Funktionsgleichung

2 Aufgabenformate und Fragentypen

Ordnen Sie die folgenden Funktionsgleichungen den skizzierten Funktionsgraphen zu:



	(A)	(B)	(C)	(D)
$f(x) = (x - 3)^2 - 3$	☐	☐	☐	☐
$g(x) = -(x + 5)^2 + 5$	☐	☐	☐	☐
$h(x) = 1/(x - 1) + 4$	☐	☐	☐	☐
$k(x) = 2x^3 + (1/2)x^2 - 2$	☐	☐	☐	☐

Hinweis **Lösung anzeigen** **Einreichen**

Abbildung 2.9: Screenshot einer Aufgabe, in der jedem Graphen die zugehörige Funktionsgleichung zugeordnet wird.

```
<exercise type="scList">
  <task>Ordnen Sie die folgenden Funktionsgleichungen den skizzierten
    Funktionsgraphen zu:
    [hier kommt der Code für den Graphen]
  </task>
  <questionlist count="row" feedbacktype="auto"
    columnNames="(A) , (B) , (C) , (D) " firstColumn="">
    <question assertion="$f(x)=(x-3)^2-3$"/>
    <question assertion="$g(x)=-(x+5)^2+5$"/>
    <question assertion="$h(x)=1/(x-1)+4$"/>
    <question assertion="$k(x)=2x^3+(1/2)x^2-2$"/>
  </questionlist>
  <correctanswer>
    <choice pattern="0001,0100,1000,0010"/>
    <message>Richtig.</message>
  </correctanswer>
  <feedback>
    <choice pattern="andere">Leider falsch.</choice>
  </feedback>
  <advice>
    <option>Kein Hinweis vorhanden.</option>
  </advice>
  <output/>
</exercise>
```

Abbildung 2.10: Beispiel für den XML-Code, um die SCList Aufgabe aus Abbildung 2.9 darstellen zu können. Um den Code lesbarer zu gestalten wurde der Teil für die Darstellung der Graphen hier weggelassen.

genau ein Graph in der Abbildung zugeordnet. Dadurch ist in jeder Zeile genau eine Antwort richtig.

Diese Tabellen-Aufgabe lassen sich nicht nur mit Checkboxes darstellen, sondern auch mit Lückentext. Damit das Stellen dieser Aufgabe einfacher wird, gibt es einen Tabellengenerator, mit dem eine Excel-Tabelle eingelesen werden kann. Für das Feedback gibt es zusätzlich, zu der Möglichkeit manuelles Feedback zu schreiben, die Möglichkeit dies durch Regeln zu automatisieren. Die richtige Lösung wird durch eine Menge von Regeln ausgezeichnet und das Feedback durch Regelverletzungen bestimmt. Für die Regeln gibt es bestimmte Voraussetzungen, die eingehalten werden müssen. Der Regelinhalt muss ein Ausdruck sein, den der Evaluator zu wahr oder falsch auswerten kann. Dabei muss nicht für jedes Kästchen eine eigene Regel geschrieben werden, manchmal reicht es eine Regel für eine Zeile oder eine Spalte zu definieren und das Feedback darauf auszurichten. Eine eingereichte Lösung wird dann als richtig vom System erkannt, wenn alle Regel erfüllt sind. Andernfalls werden die verletzten Regeln gespeichert und können im Feedback referenziert werden. Über das Attribut *test* des Tags *ruleviolation* kann gesteuert werden, wie die Regeln für das Feedback zu prüfen sind. Ist der eingetragene Wert wahr, so wird das angegebene Feedback zurückgegeben. Mögliche Werte für das Attribut sind:

2 Aufgabenformate und Fragentypen

- **exactly** Benötigt als Parameter einen Integer-Wert und gibt wahr zurück, wenn die Anzahl der verletzten Regeln diesem Wert entsprechen. Z.B. gibt *exactly(3)* wahr zurück, wenn genau 3 Regeln verletzt sind.
- **atleast** Funktioniert ähnlich wie *exactly*, nur dass wahr zurückgegeben wird, wenn die Zahl der verletzten Regeln größer oder gleich dem Parameter ist.
- **combined** Benötigt als Parameter die IDs der verletzten Regeln, getrennt durch ein Komma. Gibt wahr zurück, wenn genau diese Regeln verletzt sind. Z.B. gibt *combined(2,5,7)* wahr zurück, wenn die Regeln 2, 5 und 7 verletzt sind.
- **condition** Benötigt als Parameter einen logischen Ausdruck. Erlaubte logische Operationen sind $\&\&$ (UND), $\parallel$ (ODER) und $!$ (NICHT). Z.B. ist *condition(1 $\parallel$ 3 $\&\&$ 4)* wahr, falls Regel 1 oder 3 und 4 verletzt sind.

Abbildung 2.11 zeigt ein Beispiel für eine Tabellen Aufgabe mit Lückentext. Hier soll eine Wahrheitstafel erstellt werden für die Implikation. Durch die drei vorausgefüllten Felder ist die Tabelle eindeutig zu lösen. In der Abbildung 2.12 sind die Regeln abzulesen für die richtige Lösung der Aufgabe. In der ersten Regel wird zunächst geprüft, ob in jedem Kästchen ein „w“ oder ein „f“ steht. Eine andere Einreichung macht hier keinen Sinn. In der zweiten Regel wird geprüft, ob die ersten beiden Spalten korrekt ausgefüllt wurden. Bei den Wahrheitstafeln sind diese beiden Spalten ausschlaggebend für den Rest der Tabelle. Wenn diese schon falsch sind, kann der Rest nicht mehr korrekt ausgefüllt werden. Zum Schluss werden die Kästchen der letzten beiden Spalten auf Korrektheit überprüft. Somit wurden die Regeln für die Überprüfung der richtigen Lösung gesetzt.

Das Feedback wird nun durch die Regelverletzungen gesetzt. Das Feedback für das Beispiel der Wahrheitstafel ist in Abbildung 2.13 zu sehen. Zunächst wird getestet, ob in jedem Kästchen ein „w“ oder ein „f“ steht. Danach wird getestet, ob in den ersten beiden Spalten ein Fehler ist und ob in den letzten beiden Spalten ein Fehler ist, wenn die ersten beiden Spalten richtig sind. Nun wird Zeilenweise nach Fehlern gesucht. Zum Schluss wird die Äquivalenz in den letzten beiden Spalten getestet. Damit hat man alle Bereiche soweit abgedeckt.

Fülle die leeren Kästchen der Tabelle (Wahrheitstafel) entweder mit einem "w" (falls die betreffende Aussage wahr ist) oder mit einem "f" (falls die betreffende Aussage falsch ist) aus.

A	B	$A \Rightarrow B$	$\neg B \Rightarrow \neg A$
	f		
w			
		f	

Was kannst du anhand der Wahrheitstafel schließen?

Abbildung 2.11: Hier soll eine Wahrheitstafel erstellt werden. Als Aufgabe dient dazu eine Tabellen-Aufgabe mit Lückentext.

```
<correctanswer>
  <rule id="1">(equalsExpr(' [pos=1]', 'f') || equalsExpr(' [pos=1]', 'w'))
    &&(equalsExpr(' [pos=2]', 'f') || equalsExpr(' [pos=2]', 'w'))
    &&(equalsExpr(' [pos=3]', 'f') || equalsExpr(' [pos=3]', 'w'))
    &&(equalsExpr(' [pos=4]', 'f') || equalsExpr(' [pos=4]', 'w'))
    &&(equalsExpr(' [pos=5]', 'f') || equalsExpr(' [pos=5]', 'w'))
    &&(equalsExpr(' [pos=6]', 'f') || equalsExpr(' [pos=6]', 'w'))
    &&(equalsExpr(' [pos=7]', 'f') || equalsExpr(' [pos=7]', 'w'))
    &&(equalsExpr(' [pos=8]', 'f') || equalsExpr(' [pos=8]', 'w'))
    &&(equalsExpr(' [pos=9]', 'f') || equalsExpr(' [pos=9]', 'w'))
    &&(equalsExpr(' [pos=10]', 'f') || equalsExpr(' [pos=10]', 'w'))
    &&(equalsExpr(' [pos=11]', 'f') || equalsExpr(' [pos=11]', 'w'))
    &&(equalsExpr(' [pos=12]', 'f') || equalsExpr(' [pos=12]', 'w'))
    &&(equalsExpr(' [pos=13]', 'f') || equalsExpr(' [pos=13]', 'w'))</rule>
  <rule id="2">equalsExpr(' [pos=1]', 'f') && equalsExpr(' [pos=4]', 'f')
    && equalsExpr(' [pos=5]', 'w') && equalsExpr(' [pos=8]', 'w')
    && equalsExpr(' [pos=11]', 'w') && equalsExpr(' [pos=12]', 'f')</rule>
  <rule id="3">equalsExpr(' [pos=2]', 'w')</rule>
  <rule id="4">equalsExpr(' [pos=3]', 'w')</rule>
  <rule id="5">equalsExpr(' [pos=6]', 'w')</rule>
  <rule id="6">equalsExpr(' [pos=7]', 'f')</rule>
  <rule id="7">equalsExpr(' [pos=9]', 'w')</rule>
  <rule id="8">equalsExpr(' [pos=10]', 'w')</rule>
  <rule id="9">equalsExpr(' [pos=13]', 'w')</rule>
  <message>Richtig. Sehr gut!</message>
</correctanswer>
```

Abbildung 2.12: Dies ist ein Beispiel für den XML-Code für die Regelformulierung bei einer Tabellen-Aufgabe mit Lückentext.

2 Aufgabenformate und Fragentypen

```
<feedback>
  <ruleviolation penalty="100" test="condition(1)">Es wurde nicht jedes
    Kästchen mit einem "w" oder "f" ausgefüllt.</ruleviolation>
  <ruleviolation penalty="100" test="condition(!1&&2)">In den ersten
    beiden Spalten ist ein Fehler. </ruleviolation>
  <ruleviolation penalty="20" test="condition(!1&&2&&3||!1&&2&&4
    ||!1&&2&&5||!1&&2&&6||!1&&2&&7||!1&&2&&8||!1&&2&&9)">
    Die ersten beiden Spalten sind richtig. Bei den letzten beiden
    Spalten ist ein Fehler.</ruleviolation>
  <ruleviolation test="condition(!1&&2&&3&&4)">In der ersten Zeile ist
    ein Fehler.</ruleviolation>
  <ruleviolation test="condition(!1&&2&&3&&4)">In der ersten Zeile ist
    ein Fehler.</ruleviolation>
  <ruleviolation test="condition(!1&&2&&5&&6)">In der zweiten Zeile ist
    ein Fehler.</ruleviolation>
  <ruleviolation test="condition(!1&&2&&5&&6)">In der zweiten Zeile ist
    ein Fehler.</ruleviolation>
  <ruleviolation test="condition(!1&&2&&5&&6)">In der zweiten Zeile ist
    ein Fehler.</ruleviolation>
  <ruleviolation test="condition(!1&&2&&7&&8)">In der dritten Zeile ist
    ein Fehler.</ruleviolation>
  <ruleviolation test="condition(!1&&2&&7&&8)">In der dritten Zeile ist
    ein Fehler.</ruleviolation>
  <ruleviolation test="condition(!1&&2&&9)">In der vierten Zeile ist
    ein Fehler.</ruleviolation>
  <ruleviolation test="condition(!1&&2&&3&&8)">Denke an die Äquivalenz
    der letzten beiden Spalten.</ruleviolation>
  <ruleviolation test="condition(!1&&2&&3&&8)">Denke an die Äquivalenz
    der letzten beiden Spalten.</ruleviolation>
  <ruleviolation test="condition(!1&&2&&4&&7)">Denke an die Äquivalenz
    der letzten beiden Spalten.</ruleviolation>
  <ruleviolation test="condition(!1&&2&&4&&7)">Denke an die Äquivalenz
    der letzten beiden Spalten.</ruleviolation>
  <ruleviolation test="andere">Leider falsch.</ruleviolation>
</feedback>
```

Abbildung 2.13: Hier ist ein Beispiel für die Regelverletzungen für die Wahrheitstabelle.

3 Technische Weiterentwicklung

Nach der grundlegenden Einrichtung automatischer Prüfverfahren für Multiple-Choice-Aufgaben und Fill-In-Aufgaben sowie der zugehörigen Infrastruktur für die Verwaltung von Aufgaben und Kursen im ersten Projektjahr, lag im zweiten Jahr der Fokus der technischen Weiterentwicklung auf Funktionen im Detail. Neben dem Abschluss der bereits begonnen Integration mit dem Lern-Management-System MOODLE bedeutete dies insbesondere den Einbau eines Formeleditors und den Anschluss eines Computer-Algebra-Systems. Die dazu nötigen technischen Erweiterungen werden im Folgenden beschrieben.

3.1 Formeleditor

Da ein besonderer Fokus von JACK auf Mathematikaufgaben liegt und die Eingabe von Formeln ein wichtiger Bestandteil von solchen Aufgaben ist, wurde in JACK im zweiten Projektjahr ein Formeleditor integriert, welcher den Lernenden die Eingabe von Formeln erleichtert. So müssen die Lernenden bei der Eingabe von besonderen Zeichen, wie z.B. Wurzeln oder Potenzen nicht einer vorgegebenen Syntax folgen, sondern können diese dank eines WYSIWYG-Editors unmittelbar selbst prüfen.

Nach einer kurzen Recherche wurde eine Reihe von Formeleditoren ausgewählt, die für die Nutzung mit JACK intensiver evaluiert wurden. Besonderes Augenmerk wurde dabei auf die folgenden Punkte gelegt:

- freie Verfügbarkeit
- Kompatibilität mit den gängigsten Browsern: Firefox, Internet Explorer, Safari, Chrome, Opera
- Projekt wird noch weiterentwickelt
- große Zeichenvielfalt/Erweiterbarkeit bei mathematischen Zeichen

Die Wahl fiel schlussendlich auf den Formeleditor von MATH DOX¹. Dieser ist unter einer Open Source Lizenz verfügbar und in JavaScript implementiert, wodurch er sich einfach in Webseiten integrieren lässt. Gibt der Lernende einen mathematischen Ausdruck mit Hilfe des

¹<http://mathdox.org/formulaeditor/>

3 Technische Weiterentwicklung

Vereinfache die folgenden Ausdrücke soweit es geht und fasse sie nach Möglichkeit zusammen

$$\sqrt[4]{y^{-3}} : \sqrt[6]{y^{-5}} = \sqrt{y^2}$$

$$\sqrt[4]{\sqrt[5]{a^{20}}} = \boxed{}$$
$$\left(\frac{81a^4}{b^8}\right)^{\frac{1}{6}} : \left(\frac{b^8}{3a^4}\right)^{\frac{1}{3}} = \boxed{}$$

Abbildung 3.1: Ansicht einer Aufgabe mit Formeleditor

Formeleditors ein, so wandelt dieser den Ausdruck in OPENMATH um, welches dann intern weiterverwendet werden kann (siehe dazu auch Abschnitt 3.3).

Der Editor befindet sich noch in der Weiterentwicklung und die Standardpalette bietet eine recht große Vielfalt an mathematischen Ausdrücken. Des Weiteren können auch eigene Paletten definiert werden, wodurch sich die Menge der angezeigten Zeichen und Formeln reduzieren lässt und somit übersichtlicher für den Lernenden wird. Die bisher in JACK realisierten Paletten sind die folgenden:

- **basic:** Dieser Formeleditor besitzt nur einige grundlegende mathematische Formeln, nämlich Potenzen, Wurzeln, Brüche und Beträge. Er ist für die meisten Aufgaben ausreichend.
- **full:** Erweitert den Formeleditor „basic“ um Relationszeichen, e -Funktion und Logarithmusfunktion, Sinus und Kosinus sowie um die Symbole i (für komplexe Zahlen) und ∞ .
- **trigonometry:** Enthält nur Sinus, Kosinus, Tangens und das Symbol für π und ist somit ausschließlich für Trigonometrieaufgaben geeignet.

Ein Beispiel für den Einsatz der „basic“-Palette ist in Abbildung 3.1 gegeben. Die Menge der Paletten lässt sich für neue Themenfelder leicht erweitern. Autoren können zudem beim Erstellen einer Aufgabe konfigurieren, ob sie den Formeleditor nutzen möchten und welche Palette benötigt wird. Der Formeleditor kann dabei auch für einzelne Eingabefelder hinzu- bzw. abgeschaltet werden.

Der Formeleditor von MathDox hat jedoch auch Schwächen. So sind leider nicht alle gängigen mathematischen Ausdrücke umgesetzt, wobei sich insbesondere die Eingabe von Mengen schwierig gestaltet. Es ist jedoch zu hoffen, dass diese Ausdrücke im Rahmen der noch bestehenden Weiterentwicklung des Editors umgesetzt werden. Es ist derzeit nicht vorgesehen, den Quellcode des Editors im Rahmen des Projektes selber deutlich weiterzuentwickeln.

3.2 Anbindung an Computer-Algebra-Systeme

In E-Assessment-Systemen mit mathematischen Inhalten spielt die automatische Prüfung der vom Lernenden eingegebenen Lösungen eine wichtige Rolle. Ist die Anzahl der möglichen Lösungen endlich, wie es bei Multiple-Choice-Aufgaben der Fall ist, so hat der Autor der Aufgabe die Möglichkeit zu spezifizieren, welche Lösungen richtig und welche Lösungen falsch sind. Bei Fill-In-Aufgaben, die in JACK im ersten Projektjahr ebenfalls realisiert wurden, ist die Anzahl der möglichen Lösungen unendlich groß. Auch hier kann der Autor sich die Mühe machen und versuchen, die Menge der korrekten Eingaben zu spezifizieren, jedoch kann die Zahl der zu einer Eingabe äquivalenten Ausdrücke schon bei einfachen mathematischen Termen sehr groß werden: Der Ausdruck $2x + 2y$ lässt sich beispielsweise auch als $x + x + y + y$ oder $x + y + x + y$ darstellen. Zudem ist es auch denkbar, dass Fill-In-Aufgaben mit unendlich vielen richtigen Lösungen gestellt werden. Ein Beispiel für eine solche Aufgabe ist die Angabe einer Gerade durch einen vorgegebenen Punkt.

Die Prüfung der Eingabe des Lernenden lässt sich in solchen Fällen nur mit einem Computer-Algebra-System (CAS) bewerkstelligen, welches selbst rechnen und die Äquivalenz von mathematischen Ausdrücken überprüfen kann. Beispiele für bekannte proprietäre Computer-Algebra-Systeme sind MATHEMATICA², MAPLE³ oder MAGMA⁴. Es gibt jedoch auch eine Reihe von freien Computer-Algebra-Systemen, wie z.B. GEOGEBRA⁵ oder SAGE⁶.

Für JACK wurde nach eingehender Evaluierung das System SYMJA⁷ ausgewählt. Es ist in Java implementiert und frei verfügbar. Somit lässt es sich einfach als Bibliothek in den Code von JACK einbinden. Zudem besitzt das System einen guten Parser, der aus einem lesbaren Ausdruck einen abstrakten Syntaxbaum (AST) macht, welcher der SYMJA-Syntax entspricht. So wird etwa der Ausdruck $2x$ zu `Times[2, x]` geparkt. Daran lässt sich bereits erkennen, dass auch das „unsichtbare Multiplikationszeichen“ vom System erkannt wird. Die oft verwendete Schreibweise $2x$, die eigentlich für den Ausdruck $2 \cdot x$ steht, wird von Lernenden häufig verwendet.

Des Weiteren kann SYMJA mit Variablen rechnen und Ausdrücke, in denen Variablen vorkommen, auf einfache aber effiziente Weise miteinander vergleichen. Allerdings haben sich im praktischen Einsatz auch Nachteile dieser Lösung gezeigt: Schwächen besitzt SYMJA beispielsweise auf dem Gebiet der Linearen Algebra, da es für diesen Bereich nur sehr wenige Funktionen anbietet, die z.T. auch ungenau arbeiten. So konnte SYMJA in einem Testfall die

²<http://www.wolfram.com/mathematica/>

³<http://www.maplesoft.com>

⁴<http://magma.maths.usyd.edu.au/magma/>

⁵<http://www.geogebra.org>

⁶<http://www.sagemath.org>

⁷<http://code.google.com/p/symja/>

Eigenwerte der Matrix

$$\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

nicht exakt berechnen. Zudem fehlt die Implementation der Areafunktionen und Mengenbeziehungen. Ein weiterer Nachteil von SYMJA ist die mangelnde Dokumentation der Anwendung. Es gibt zwar eine Projektseite, jedoch fehlen Erläuterungen zu den einzelnen Funktionen des Systems völlig. Es ist lediglich eine Seite mit einer Reihe von Testfällen verfügbar. Dadurch gestaltet es sich schwierig, sämtliche Funktionen zu verstehen.

Als Reaktion auf diesen Mangel wurde im Laufe des zweiten Projektjahres mit der Erstellung einer eigenen Dokumentation begonnen. Im dritten Projektjahr sollen jedoch auch noch einmal Alternativen geprüft werden, um diese statt SYMJA in JACK zu integrieren.

3.3 OPENMATH

Aus der Anbindung von JACK an den MathDox-Formeleditor ergaben sich weitere Anforderungen, welche die Speicherung von Lösungen betreffen. Während die Eingabe bei Aufgaben ohne Formeleditor zuvor intern direkt gespeichert wurde, wandelt der Formeleditor diese in das OPENMATH-Format um. OPENMATH⁸ ist ein XML-basierter Standard zur Beschreibung der Semantik von mathematischen Ausdrücken, welcher dazu dient mathematische Ausdrücke maschinenlesbar zu machen, um sie in Software-Anwendungen benutzen zu können. Der Standard definiert dafür OPENMATH-Objekte, einen abstrakten Datentyp, welcher dazu dient, die funktionale Struktur eines mathematischen Ausdrucks zu beschreiben. OPENMATH-Objekte können aus Symbolen (OPENMATH-Symbole), Variablen (OPENMATH-Variablen), Funktionen (OPENMATH-Funktionen) und Bindungsobjekten (OPENMATH-Binding Objects) bestehen. Die Bedeutung eines OPENMATH-Symbols wird in einem „Content Dictionary“ (CD) definiert, einer externen Datei, welche vom jeweiligen Symbol zu referenzieren ist. Alle CDs sind auf der offiziellen OPENMATH-Webseite verfügbar. Insgesamt listet der Standard dort 200 CDs auf, von denen 36 offiziellen Status besitzen. Die CDs können von jedem Nutzer des Standards erweitert werden und es können auch eigene CDs ergänzt werden. Ein Beispiel für die Umsetzung eines mathematischen Ausdrucks in OPENMATH ist in Listing 3.2 gegeben: Der Ausdruck $3 + 5$ wird in der XML-Struktur in Teilausdrücke zerlegt, die die Werte 3 bzw. 5 sowie die Verknüpfung durch die Addition repräsentieren.

Die Semantik eines im OPENMATH-Format geschriebenen mathematischen Ausdrucks lässt sich somit gut verstehen. Das ist insbesondere dann wichtig, wenn man diesen auf seine Korrektheit überprüfen möchte. Diese Aufgabe übernimmt jedoch, wie in Kapitel 3.2 beschrieben,

⁸<http://www.openmath.org>

```
<OMOBJ>
  <OMA>
    <OMS cd="arith1" name="plus"/>
    <OMI>3</OMI>
    <OMI>5</OMI>
  </OMA>
</OMOBJ>
```

Abbildung 3.2: Der Ausdruck $3 + 5$ in der XML-Struktur von OPENMATH.

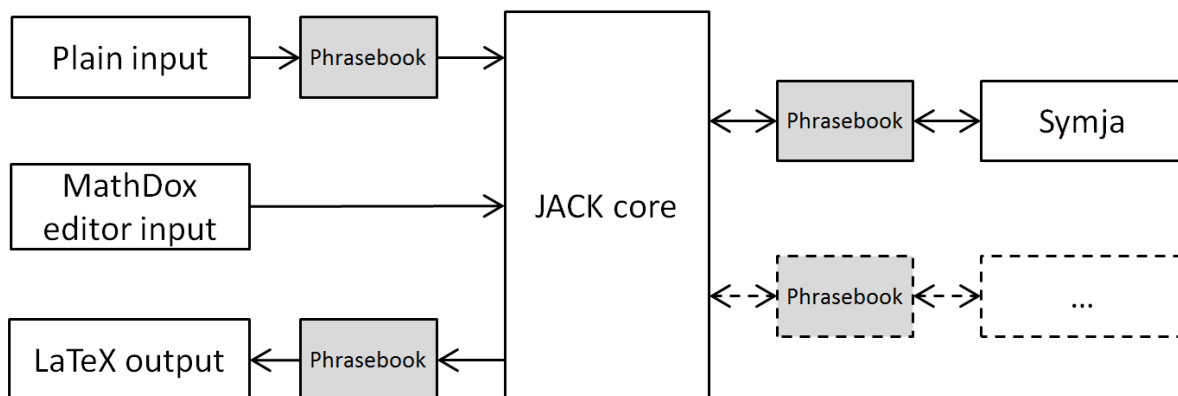


Abbildung 3.3: Architektur von JACK bezüglich der Handhabung mathematischer Eingaben. Während der verwendete Formeleditor bereits Daten im OPENMATH-Format ausliefert, wird für die Übersetzung anderer Eingaben ein Phrasebook benötigt. Jegliche Kommunikation mit einem CAS muss ebenfalls in beide Richtungen durch ein Phrasebook übersetzt werden. Für die Ausgabe ist schließlich ein weiteres Phrasebook nötig, dass von OPENMATH nach $\text{\LaTeX}$ übersetzen kann.

ein CAS. Dieses benötigt den Ausdruck jedoch in der Regel in seiner eigenen Syntax, d.h. im Fall von JACK in der Syntax von SYMJA. Daher ist ein weiterer Dienst notwendig, welcher die Syntax des CAS und OPENMATH kennt und zwischen beiden übersetzen kann. Eine Software, welche diese Aufgabe übernimmt, wird vom Standard als *Phrasebook* bezeichnet.

In JACK werden gleich an mehreren Stellen Phrasebooks benötigt. Da die Speicherung der Lösungen einheitlich geschehen soll, werden diese von nun an ausschließlich im OPENMATH-Format gespeichert. Somit muss zunächst die Eingabesyntax in OPENMATH übersetzt werden, damit die Lösung in der Datenbank gespeichert werden kann. Da der oben diskutierte Formeleditor Eingaben direkt in das OPENMATH-Format umwandelt, ist hier keine Übersetzung notwendig. Zur Überprüfung der Eingabe wird jedoch SYMJA herangezogen. Dafür muss die im OPENMATH-Format gespeicherte Eingabe in die von SYMJA benötigte Syntax übersetzt werden. Im Anschluss an die Überprüfung erhalten die Lernenden Feedback, in welchem auch $\text{\LaTeX}$ -Code angezeigt werden kann. Damit man im Feedbacktext auf die Einreichung des Lernenden eingehen kann, ist es folglich notwendig, die OPENMATH-Darstellung der Eingabe in eine $\text{\LaTeX}$ -Darstellung übersetzen zu können. Die so entstehende Architektur der Lösungsüberprüfung kann in [Abbildung 3.3](#) betrachtet werden.

Im zweiten Projektjahr wurden bisher zwei Phrasebooks realisiert: Ein SYMJA-Phrasebook, welches von OPENMATH in die SYMJA-Syntax und zurück übersetzen kann und ein $\text{\LaTeX}$ -Phrasebook, welches nur von OPENMATH nach $\text{\LaTeX}$ übersetzt. Beide Phrasebooks wurden erfolgreich in JACK integriert und werden zur Lösungsüberprüfung und Erzeugung des Feedbacks herangezogen.

Durch die oben beschriebene Architektur lässt sich JACK leicht um weitere CAS erweitern, indem die für die Übersetzung nötigen Phrasebooks erstellt werden. Dies ist vor allem vor dem Hintergrund der in Kapitel 3.2 beschriebenen Schwächen von SYMJA sinnvoll. Zudem kann JACK auch um Systeme ergänzt werden, welche von SYMJA noch nicht abgedeckte Teilbereiche der Mathematik repräsentieren können.

3.4 MOODLE-Integration

Der erste Teil der Anbindung von JACK an MOODLE 2⁹ wurde im ersten Projektjahr realisiert. Diese Integration entspricht dem IMS-LTI-Standard¹⁰. Die Studierenden können dadurch von MOODLE 2 aus direkt zu einer Aufgabe bzw. zu einem Kurs in JACK weitergeleitet werden. Lehrende brauchen dazu in MOODLE 2 lediglich eine Aktivität vom Typ „externes Tool“ anzulegen und dort u.a. die ID der Aufgabe bzw. des Kurses in JACK anzugeben. Wenn sich die Studierenden dann in MOODLE 2 einloggen, können sie auf diesen Link klicken. Im selben Fenster erscheint dann direkt die Aufgabe bzw. der Kurs in JACK.

Im zweiten Projektjahr wurde die Kommunikation zwischen MOODLE 2 und JACK dahingehend erweitert, dass nun die Ergebnisse der Studierenden in JACK an MOODLE 2 weitergeleitet werden. Auch dieser Teil der Kommunikation ist über den IMS-LTI-Standard definiert. Wenn Studierende in JACK eine Lösung einreichen wird die erreichte Punktzahl automatisch an MOODLE 2 weitergeleitet, sofern die Studierenden zuvor von MOODLE 2 aus zu JACK geleitet worden waren. Für Studierende, die JACK direkt aufrufen, werden auch keine Informationen an MOODLE 2 geleitet. Für die Weiterleitung der Punkte wird jedesmal ein spezielles XML-Dokument erstellt und als XMLHttpRequest-Objekt an Moodle2 verschickt. Das XML Dokument erhält neben den erreichten Punkten, die User-ID der Studierenden sowie weitere Informationen damit Moodle2 dies über den OAuth-Protokoll¹¹ verifizieren kann.

Durch diese Rückkopplung können beispielsweise die Lernpfade in MOODLE 2 nun auch bei der Nutzung von JACK genutzt werden. Sowohl die Lehrenden als auch die Studierenden können zudem in der Bewertungsübersicht in MOODLE 2 die jeweiligen erreichten Punkte einsehen (siehe Abbildungen 3.4 und 3.5. Die Studierenden sehen hierbei ihre eigenen Ergebnisse und die Lehrenden die Ergebnisse aller Studierenden im Kurs. Es ist zu beachten, dass nur

⁹<https://moodle.org/>

¹⁰<http://www.imsglobal.org/lti/index.html>

¹¹<http://oauth.net/>

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

Kategorie	Thema	Gruppe	Erreichte Punkte	Mögliche Punkte	Prozent
Kurs: Thema mit mehreren Variablen	Gruppe 1	21,85	70,00	31,25	44,21 %
	Gruppe 2	1,79	5,94	33,00	6,05 %
Kurs: Lineare Funktionen	Gruppe 1	60,89	6,88	7,14	88,08 %
	Gruppe 2	-	-	-	-
Kurs: Systeme linearer Gleichungen	Gruppe 1	-	-	-	-
	Gruppe 2	-	-	-	-
Kurs: Quadratische Funktionen und ihr Graph	Gruppe 1	-	-	-	-
	Gruppe 2	-	-	-	-
Kurs: Quadratische Gleichungen	Gruppe 1	43,39	38,29	2,71	113,01 %
	Gruppe 2	7,89	17,89	84,00	9,27 %
Kurs: Flächeninhalte am Dreieck	Gruppe 1	6,89	-	2,29	300,00 %
	Gruppe 2	-	-	-	-
Kurs: Potenzieren	Gruppe 1	61,35	6,88	8,87	78,07 %
	Gruppe 2	-	-	-	-

Abbildung 3.4: Dies ist die Ansicht der Bewertungsübersicht für die Lehrenden in MOODLE 2. Die Punkte in den jeweiligen Kursen, die die Lernenden in JACK erreicht haben, werden hier aufgelistet.

Bewertungsskala	Bewertung	Erreichte Punkte	Prozent	Feedback
Kurs: Thema mit mehreren Variablen	20,00	0-100	20,00 %	
Kurs: Thema mit mehreren Variablen	7,84	0-100	7,84 %	
Kurs: Lineare Funktionen	11,89	0-100	11,89 %	
Kurs: Systeme linearer Gleichungen	5,87	0-100	5,87 %	
Test 1	-	0-100	-	
Kurs: Bruchterme	10,91	0-100	10,91 %	
Kurs: Quadratische Funktionen und ihr Graph	-	0-100	-	
Kurs: Quadratische Gleichungen	6,00	0-100	6,00 %	
Kurs: Flächeninhalte am Dreieck	-	0-100	-	
Kurs: Potenzieren	-	0-100	-	
Kurs: Exponentialfunktionen	-	0-100	-	
Test 2	-	0-100	-	
Test 3	-	0-100	-	
Summe für den Kurs	6,00	0-100	6,00 %	

Abbildung 3.5: Dies ist die Bewertungsübersicht für die Lernenden in MOODLE 2. Je nach Kurs werden die erreichten Punkte dargestellt. Die Kurse, die nicht bearbeitet wurden enthalten keine Punkte.

die letzten erreichten Punkte in MOODLE 2 angezeigt werden, während die vorher erreichten Punkte mit der Einreichung einer neuen Lösung derselben Aufgabe immer überschrieben werden.

4 Praktischer Einsatz

Im zweiten Projektjahr wurde der praktische Einsatz von JACK in Lehrveranstaltungen erweitert. Bisher war der Einsatz nur auf Java Programmierung beschränkt seit 2013 werden auch mathematiklastige Veranstaltungen begleitet. In den vergangenen Sommermonaten fand bspw. ein Mathematik Vorkurs statt, der mithilfe von JACK durchgeführt wurde. Die folgende Tabelle gibt ein Überblick über die Lehrveranstaltungen bei denen JACK im Einsatz war und durch die positive Erfahrung auch weiter eingesetzt wird.

4.1 Erfahrungen aus dem vorlesungsbegleitenden Einsatz

JACK wird bereits erfolgreich in mehreren Veranstaltungen vorlesungsbegleitend genutzt. Da sich die Aufgabenstellung im Moment noch etwas komplizierter darstellt wurde zunächst mit den betreffenden Mitarbeitern der Lehrstühle ein Workshop veranstaltet. In diesem Workshop wurde anhand von Beispielen und Selbstversuchen erklärt, wie diese xml-Dateien zu erstellen sind. Anschließend übernahmen die Mitarbeiter der zugehörigen Lehrstühle die Erstellung ihrer Aufgaben.

4.1.1 Sommersemester 2013

Im Sommersemester 2013 wurde JACK zunächst in der Veranstaltung Mikroökonomie I von Professor Amann vorlesungsbegleitend eingesetzt. Jede Woche wurden Übungsaufgaben in JACK gestellt, so dass im Verlauf des Semesters insgesamt 30 Übungsaufgaben angeboten wurden. Den Studierenden wurde dadurch die Möglichkeit gegeben, den Vorlesungsstoff zu

Semester	Veranstaltungen	Teilnehmer	Größe des Aufgabenpools	Testat Aufgaben
SS 2012	Vorkurs Mathematik	131	169	10
SS 2013	Mikroökonomie I	722	28	17
	Vorkurs Mathematik	177	170	30
WS 2013/14	Mikroökonomie I	415	52	15
	Mikroökonomie II	604	12	11
	Deskriptive Statistik	657	93	23

Tabelle 4.1: Liste der Veranstaltungen mit Teilnehmern und Größe des Aufgabenpools

üben und zur Lösung der Aufgaben praktisch anzuwenden. Der Vorteil gegenüber klassischen Übungszetteln besteht darin, dass die Eingabe sofort von dem System korrigiert wird. Eine richtige Lösung wird als richtig deklariert und bei einer falschen wird ein passendes Feedback generiert. Durch die Möglichkeit der variablen Aufgabenstellung ändern sich die Werte bei erneuter Bearbeitung. Die Aufgaben konnten so mehrmals bearbeitet werden. Um die Studierenden schon früh im Semester dazu zu bringen sich mit dem Vorlesungsstoff auseinanderzusetzen wurden zudem 5 Testate mit insgesamt 10 Aufgaben angeboten, durch die die Studierenden Zusatzpunkte für die Klausur sammeln konnten. Auch hier erwiesen sich die variablen Aufgabenstellungen als vorteilhaft: Die Studierenden konnten diese Testate ohne Aufsicht bearbeiten, denn jeder Teilnehmer bekommt dieselbe Aufgabe mit anderen Werten. Im Unterschied zu den Übungsaufgaben durfte ein Testat allerdings nur jeweils einmal bearbeitet werden und die Bearbeitungszeit war zeitlich beschränkt.

Insgesamt schrieben ca. 600 Studierende am Ende des Semesters die Klausur mit. In einem Vergleich mit den Vorjahren, in denen JACK noch nicht im Einsatz war, ist die Abschlussklausur im Sommersemester 2013 erheblich besser ausgefallen. Während im Sommersemester 2011 lediglich 47% die Klausur bestanden haben waren es im Sommersemester 2013 76%. Die Bestehensrate hat sich also fast verdoppelt. Es liegen leider keine Zahlen für das Sommersemester 2012 vor, da die Veranstaltung in diesem Jahr von einem anderen Dozenten gehalten wurde.

4.1.2 Wintersemester 2013/14

Im Wintersemester 2013/2014 ist der Einsatz von JACK auf die Vorlesungen Mikroökonomie II von Professor Amann und Deskriptive Statistik I von Professor Hanck erweitert worden. Beide Veranstaltungen übernahmen das Konzept aus dem Sommersemester 2013 mit wöchentlichen Übungsaufgaben und 5 Testaten über das Semester verteilt. Die 12 Aufgaben der Vorlesung Mikroökonomie II wurden von 604 Studierenden bearbeitet und die 93 Aufgaben der Vorlesung Deskriptive Statistik I von 657 Studierenden.

Zusätzlich dazu wurde die Vorlesung Mikroökonomie I im Wintersemester 2013/2014 als reine e-Learning Veranstaltung in Moodle 2 angeboten. Anstatt der wöchentlichen Vorlesung wurden Videos online gestellt. Der Übungsbetrieb und die Testate liefen in JACK ab, wie bei den anderen Veranstaltungen auch. Dies hatte den Hintergrund, dass üblicherweise diese Veranstaltung nur alle zwei Semester und zwar immer im Sommersemester angeboten wurde. Mithilfe von JACK konnte nun diese Veranstaltung zusätzlich auch im Wintersemester angeboten werden. Die Aufgaben von Mikroökonomie I werden von ca. 200 Studierenden bearbeitet.

4.2 Erfahrungen aus den Vorkursen

Im September 2012 und 2013 wurde jeweils ein Mathematik Vorkurs für die Fakultät Wirtschaftswissenschaften angeboten. Dieser richtete sich an Studienanfänger der Fächer Betriebswirtschaftslehre, Volkswirtschaftslehre, Wirtschaftsinformatik, Systems Engineering/ Angewandte Informatik, Lehramt Wirtschaftswissenschaften und Lehramt Informatik. Das Ziel des Vorkurses besteht darin, eine Brücke zwischen der Schule und der Universität zu schaffen.

4.2.1 Veröffentlichung zum Vorkurs 2012

Bereits im ersten Projektjahr war im September 2012 ein Mathematik-Vorkurs an der Fakultät für Wirtschaftswissenschaften angeboten worden. Dabei konnte ein deutlicher Motivationsschwund der Studierenden im Verlauf des dreiwöchigen Kurses festgestellt werden. Um Gegenmaßnahmen entwerfen zu können wurde dieses Phänomen zu Beginn des zweiten Projektjahres genauer untersucht. Über die Ergebnisse ist ein Paper auf der DeLFI 2013 in Bremen veröffentlicht worden [SKKS⁺13]. In diesem Paper wurde der Zusammenhang zwischen der Motivation der Studierenden und verschiedenen Frageformaten untersucht. Dazu ist zunächst eine Auflistung der Frageformate vorgenommen worden, die in dem Vorkurs vorkamen. Im gleichen Schritt wurde auch die Anzahl der jeweiligen Frageformate, die Anzahl der Lösungen und die Anzahl der Studierenden, die die jeweiligen Aufgaben gelöst haben, gezählt. Dabei ist aufgefallen, dass ein Unterschied bei der Bearbeitung der jeweiligen Aufgabe zu erkennen ist. Zudem gibt es in der Literatur eine Unterteilung in konvergente und divergente Aufgaben und in offene und geschlossene Aufgaben. Bei konvergenten Aufgaben ist die Lösungsmenge genau definiert und bei divergenten Aufgaben gibt es beliebig viele richtige Lösungen. Es handelt sich um eine geschlossene Aufgabe, wenn die Lösungsmöglichkeiten vorgegeben sind, und offen, wenn die Lösung von dem Studierenden selbst eingetragen werden muss. Daraus ergeben sich die folgenden Konstellationen in Tabelle 4.2 für die Aufgaben aus dem Vorkurs.

	konvergenter Aufgabentyp	divergenter Aufgabentyp
geschlossener Aufgabentyp	Multiple-Choice Drop-Down	-
offener Aufgabentyp	Fill-In	Fill-In

Tabelle 4.2: Welcher Aufgabentyp sind die einzelnen Frageformate in JACK.

Am Ende konnte ein Unterschied zwischen geschlossenen und offenen Aufgabentypen festgestellt werden. Bei geschlossenen Aufgabentypen wurde häufiger nach einer falschen Lösung weitergerechnet, als bei offenen Aufgabentypen. Leider eignen sich die geschlossenen Aufgabentypen nur für die Wissensabfragen, in der Mathematik möchte man aber in der Regel mehr abfragen. Hier ist es auch wichtig die mathematischen Rechnungen selbständig lösen zu

können. Aus diesem Grund stellt sich die Frage, wie in Zukunft offene Aufgabentypen attraktiver gestaltet werden können, so dass auch bei dem Frageformat Fill-In mehr Einreichungen gemacht werden. Das Ziel ist, dass Studierende anhand vom Feedback bei falschen Lösungen die richtige Lösung finden können. Darum muss das Feedback immer auf die eingereichte Lösung eingehen.

4.2.2 Vorkurs 2013

Der Vorkurs 2012 wurde als Blended-Learning Kurs angeboten wurde, indem ein Präsenzanteil in Form von Vorlesungen und Übungsgruppen angeboten und durch weitere automatisierte Übungen in JACK ergänzt wurde. Abgesehen von der oben bereits diskutierten schwindenden Motivation bei den Übungen in JACK zeigten auch die Evaluationsergebnisse zu diesem Vorkurs, dass die Studierenden nur an den Übungsstunden teilgenommen haben und kaum in JACK geübt haben. Da jedoch eine angemessene und dauerhafte Präsenzbetreuung in den kommenden Jahren nicht sichergestellt ist und zudem in jedem Fall schon aus logistischen Gründen eine Beschränkung der Teilnehmerzahl bedingt, wurde eine Veränderung im zweiten Vorkurs unternommen, indem feste Präsenztermine aus dem Konzept gestrichen wurden. Stattdessen wurde über MOODLE 2 ein Skript bereitgestellt und passende Aufgaben mit einem Link zu JACK. Zusätzlich wurde zweimal pro Woche eine Sprechstunde angeboten, in denen die Studierenden technische und inhaltliche Fragen an Tutoren stellen konnten. Zudem wurde die Möglichkeit eröffnet, Fragen in einem Forum in MOODLE 2 zu stellen. Durch dieses Vorgehen musste die Teilnehmerzahl für den Vorkurs nicht begrenzt werden. tatsächlich nahmen an diesem Vorkurs zu Beginn ca. 170 Studierende teil, während im Vorjahr nur 150 Anmeldungen angenommen worden waren.

Durchgeführt wurde der Vorkurs im zweiten Projektjahr im September 2013 und hatte wie im Vorjahr eine Dauer von 3 Wochen. Der Inhalt richtete sich wie zuvor nach dem Schulstoff bis zur Klasse 10 im alten G9. Es gab ca. 170 Übungsaufgaben und am Ende jeder Woche wurde ein Testat geschrieben mit jeweils 10 Aufgaben über die vergangene Woche. Das letzte Testat war eine Abschlussklausur und behandelte alle Themengebiete.

In Tabelle 4.3 sind zum einen die Zahlen abzulesen, wie viele Studierende durchschnittlich die Aufgaben in einem der 12 Themengebiete bearbeitet haben und zum anderen wie viele Lösungen in den jeweiligen Themengebieten eingereicht wurden. In Abbildung 4.1 sind diese Zahlen grafisch dargestellt. So ist deutlicher der Abwärtstrend über die verschiedenen Themengebiete zu erkennen. In jeder Woche wurden vier Themengebiete veröffentlicht. Das erklärt vielleicht, warum das fünfte und neunte Themengebiet jeweils einen kurzen Aufwärtstrend zeigen. Obwohl es in diesem Jahr keine Präsenzveranstaltungen zusätzlich zu den Aufgaben in JACK gab, wurden die Aufgaben im letzten Themengebiet von nur noch 12,4% der ursprünglichen Anzahl Studierender bearbeitet. Der Grund für diesen Abwärtstrend könnte schlichtweg

Themengebiet	Anzahl der Studenten im Durchschnitt	Anzahl der Lösungen im Durchschnitt
1 Terme mit einer Variable	117,75	164,8
2 Terme mit mehreren Variablen	95,94	117,41
3 Lineare Funktionen	78,07	88,71
4 Systeme linearer Gleichungen	62,27	74,33
5 Bruchterme	68,64	81,82
6 Quadratwurzel	58,75	72,75
7 Quadratische Funktion und ihr Graph	46,4	55,1
8 Quadratische Gleichungen	32,07	35
9 Flächensätze am Dreieck	32,67	35,67
10 Potenzen	30,45	31,4
11 Exponentialfunktionen	23	25,09
12 Trigonometrie	14,6	15,6

Tabelle 4.3: Überblick über die Anzahl der Studierenden, die die Aufgaben in dem dreiwöchigen Vorkurs bearbeitet haben.

damit zusammenhängen, dass die Studierenden sich zunächst alle eingeloggt haben und die Aufgaben bearbeitet haben. Nach kurzer Zeit jedoch festgestellt haben, dass sie den Inhalt bereits kennen und die Aufgaben lösen können.

Abbildung 4.4 zeigt das Nutzerverhalten in den einzelnen Aufgaben des ersten Themengebietes „Terme mit einer Variable“. Abzulesen sind die Anzahl der Studierenden, die eine Aufgabe bearbeitet haben, die Anzahl der eingereichten Lösungen, die Anzahl der richtig eingereichten Lösungen und die Anzahl der falsch eingereichten Lösungen, die Anzahl der Studierenden, die die jeweilige Aufgabe mehrmals bearbeitet haben (einmal in absoluten Zahlen und einmal in Prozent). Dadurch lässt sich das Verhältnis mit den jeweiligen Aufgaben besser vergleichen. Daneben steht der Aufgabentyp und wie häufig die jeweiligen Stufen insgesamt übersprungen wurden. Hier ist nun zu beobachten, dass relativ wenige Studierende eine Aufgabe mehr als einmal bearbeitet haben. Im Schnitt haben 25,83% Studierende, also jeder vierte, eine Aufgabe mehr als einmal bearbeitet. Lediglich bei den beiden Aufgaben „Term vereinfachen und Gleichung lösen 2“ und „Terme“ haben ca. 50% aller Studierenden die Aufgabe mehr als einmal bearbeitet. Es wurde also auch in diesem Vorkurs kein besseres Ergebnis erzielt, als im vorherigen Jahr. Der Grund könnte sein, dass diesmal alle Aufgaben eines Themengebietes zusammen in einem Kurs gefasst waren. Dadurch musste zunächst der Kurs angeklickt werden

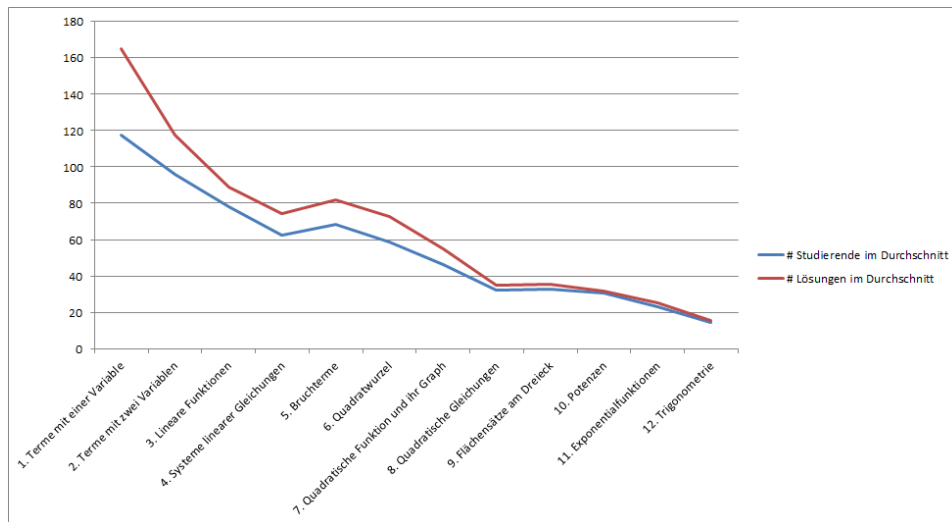


Abbildung 4.1: Graphische Darstellung über die Anzahl der Studierenden, die ein Themengebiet bearbeitet haben und die Anzahl der eingereichten Lösungen der jeweiligen Themengebiete.

und dann die jeweilige Aufgabe. Solange sich also ein Student/eine Studentin in einem Kurs aufhielt änderte sich die Aufgabe nicht, es konnte lediglich das Ergebnis verbessert werden. Da teilweise bis zu zwanzig Aufgaben in einem Kurs waren, könnte es sein, dass die Studierenden die Kurse jeweils nur einmal bearbeitet haben.

Die Frage, die am Ende noch offen bleibt ist „Kann man die Studierende in Zukunft besser motivieren, so dass sie in der letzten Woche genauso gut arbeiten, wie in der ersten?“. Vielleicht ist dies aber auch gar nicht notwendig, da die Meisten den Stoff bereits gut genug beherrschen. Bis jetzt war kein Wissen darüber bekannt wie das mathematische Vorwissen der Teilnehmer tatsächlich aussah. Von späteren Lehrveranstaltung ist bekannt, dass bei einigen Studenten Lücken sind. Das Ziel in der Zukunft soll sein durch einen Selbsteinstufungstest Aufschluss über das tatsächliche Vorwissen der Teilnehmer zu bekommen. Dadurch sollen die Lücken besser erkannt werden und ein Test am Ende des Vorkurses soll die Verbesserungen durch den Vorkurs gezeigt werden.

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	Welche der einzelnen Stufen, wie häufig übersprungen wurde
Gesetze	128	181	140	41	36	28,13%	Drop-Down	50
Löse die Betragsgleichung	86	121	55	66	26	30,23%	Mehrstufig	22 14 10
Löse die Ungleichung	101	134	93	41	24	23,76%	Mehrstufig	12 13 22 2 9
Löse die Ungleichung 1	84	134	33	101	19	22,62%	Fill-In	71
Lösungsmenge Gleichung	124	159	143	16	30	24,19%	Fill-In	14
Oberfläche eines Würfels	122	158	120	38	28	22,95%	Mehrstufig	20 2 15
Term vereinfachen und Gleichung lösen	117	150	126	24	29	24,79%	Fill-In	20
Term vereinfachen und Gleichung lösen 2	145	261	154	107	68	46,90%	Mehrstufig	4 3 10 11 10
Terme	151	350	229	121	84	55,63%	Mehrstufig	61 3
Terme aufstellen	127	167	133	34	29	22,83%	Fill-In	18
Terme aufstellen und vereinfachen	111	134	106	28	20	18,02%	Mehrstufig	1 8 6 9 16
Terme vereinfachen	137	189	160	29	41	29,93%	Fill-In	25
Terme zusammenfassen	124	164	114	50	31	25%	Fill-In	38
Termumformung	128	172	156	16	35	27,34%	Fill-In	13
Termumformungen mit Fehlern	117	137	109	28	18	15,38%	MC	25
Textaufgabe Terme	107	132	120	12	18	16,82%	Mehrstufig	5 4
Ungleichung aufstellen und lösen	105	134	131	3	24	22,86%	Mehrstufig	4 14
Waagschale	116	136	131	5	18	15,52%	Mehrstufig	9 1
Wasserhöhe im Swimmingpool	104	131	90	41	23	22,12%	Mehrstufig	9 7 4
Zahlenrätsel	121	152	140	12	26	21,49%	Mehrstufig	8 10

Tabelle 4.4: Überblick über das Nutzerverhalten im ersten Themenblock des Mathematik-Vorkurses im WS 2013/14

5 Fazit und Ausblick

5.1 Bilanz des zweiten Projektjahres

Obwohl einige Aspekte des Maßnahmenplans für das zweite Projektjahr nicht umgesetzt werden konnten, kann eine deutlich positive Bilanz gezogen werden. Dabei sticht insbesondere der umfangreiche Einsatz von JACK in mehreren Lehrveranstaltungen hervor, der für diese Projektphase nicht vorhersehbar war und auf sehr positives Echo gestoßen ist. Auch wenn die Datengrundlage für eine solide Evaluation noch viel zu dünn ist, kann aufgrund einzelner Klausurergebnisse schon ein positiver Einfluss aus dem vorlesungsbegleitenden Einsatz von JACK gezogen werden. Ebenso kann festgestellt werden, dass das bisherige technische Konzept von JACK sich als tragfähig erwiesen hat und dass es möglich war, Funktionalität und Inhalte im zweiten Projektjahr sehr zügig nach den Wünschen der Dozenten anzupassen und zu erweitern.

Durch den gestiegenen praktischen Einsatz von JACK in verschiedenen Lehrveranstaltungen sind zudem neue Herausforderungen und Ideen für das dritte Projektjahr entstanden. Einige davon, wie beispielsweise die Entwicklung weiterer Aufgabentypen, sind ohnehin im Projektplan vorgesehen gewesen, während andere den Maßnahmenplan des dritten Jahres bedarfsgerecht ergänzen werden.

5.2 Anstehende Arbeiten im drittem Projektjahr

Da nun bereits mehrere Lehrstühle und Tutoren im Einsatz von JACK geschult sind und umfangreiche Aufgabenpools zur Verfügung stehen, wird der Fokus im dritten Projektjahr wieder mehr auf der technischen Ausweitung von JACK liegen. Im Folgenden werden dazu sowohl die neuen Maßnahmen und Ideen, die während des zweiten Projektjahres entstanden sind, als auch die im allgemeinen Projektplan vorgesehenen Maßnahmen in die drei Maßnahmenpakete unterteilt und erläutert.

5.2.1 Kommende inhaltliche Arbeiten

Auf der inhaltlichen Ebene sieht der allgemeine Projektplan die Identifikation neuer Aufgabentypen sowie den Entwurf und die Pflege von Aufgaben vor. Aus dem praktischen Einsatz haben sich bereits zahlreiche Ideen für neue Aufgabentypen ergeben (z.B. Zuordnungsaufgaben, interaktive grafische Aufgaben, oder das Führen von mathematischen Beweisen), die im dritten Projektjahr analysiert und bedarfsgerecht umgesetzt werden sollen.

5.2.2 Kommende technische Arbeiten

Aus den inhaltlichen Arbeiten ergeben sich zwangsläufig die im allgemeinen Projektplan ebenfalls vorgesehen technischen Arbeiten bezüglich des Entwurfs und der Realisierung neuer Prüfkompontenten. Zu diesen Arbeiten gehört auch, den Einsatz weiterer Computer-Algebra-Systeme zu prüfen und zu realisieren, um so eine breitere fachliche Abdeckung im Bereich Mathematik zu erreichen.

Im Rahmen der ebenfalls vorgesehenen fortlaufenden Optimierung ist zudem geplant, die Systemarchitektur von JACK dahingehend zu überarbeiten, dass flexiblere Kurskonzepte und die Implementierung eines Frameworks für geführtes Lernen ermöglicht werden sollen. Insbesondere soll das Kurskonzept von JACK so erweitert werden, dass drei Arten von Kursen angeboten werden können:

- **Summativer Test**

Für diese Art von Test bzw. Kurs wählt der Lehrende eine Menge von Aufgaben und kann bei Bedarf eine max. Bearbeitungszeit für diesen Test ergänzen. Studierende können diesen Test nur einmal bearbeiten und bekommen am Ende erreichte Punkte bzw. bzw. eine Note angezeigt. Diese Art von Tests gibt es derzeit bereits in JACK als „Klausurmodus“.

- **Formativer Test**

Dieser Kurs wird aus einer Menge von Aufgaben bestehen, die in einem thematischen Zusammenhang stehen, aber wahlfrei und wiederholt bearbeitet werden können. Verschiedene Kurse können bei Bedarf auch hintereinander verkettet werden. In dieser Art von Tests soll das Framework für geführtes Lernen auf Basis eines Nutzermodells und einer Schwierigkeitsschätzung für die jeweiligen Aufgaben zum Einsatz kommen, um den Studierenden jeweils passende Aufgaben aus dem großen Aufgabenvorrat dieses Tests vorzuschlagen.

- **Diagnostischer Test**

Dieser Test ist im Prinzip genauso wie der summativer Test, nur mit dem Unterschied, dass keine Punkte bzw. Note als Ergebnis berechnet werden, sondern eine spezifische

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

Auswertung für den Lernenden formuliert wird, wie z.B. eine Empfehlung welche Bereiche der Lernende verbessern sollte. Zudem kann diese Art von Test für die Initialisierung des Nutzermodells für neue Studierende verwendet werden.

Über den Projektplan hinaus wird im dritten Projektjahr außerdem an einem Aufgabeneditor gearbeitet, der Lehrenden das Stellen von Aufgaben erleichtern soll.

5.2.3 Kommende organisatorische Arbeiten

Der vorlesungsbegleitende Einsatz von JACK wird voraussichtlich auch im dritten Projektjahr im selben Umfang wie im zweiten Projektjahr stattfinden. Zudem ist vorgesehen, nicht nur im September 2014, sondern bereits im März 2014 einen Mathematik-Vorkurs anzubieten. Weitere Einsatzmöglichkeiten von JACK in anderen Lehrveranstaltungen werden derzeit mit verschiedenen Lehrstühlen besprochen. Zudem ist vorgesehen, im Rahmen eines Workshops an der Fakultät für Wirtschaftswissenschaften über die bisherigen Einsatzszenarien zu berichten.

Um den Studierenden den Umgang mit JACK zu erleichtern, soll im dritten Projektjahr zudem eine Reihe von Screencasts produziert werden, die die verschiedenen Funktionen und Aufgabentypen von JACK zielgruppengerecht erläutern und so in die Nutzung einführen.

Ein weiterer wichtiger organisatorischer Punkt ist die Sicherstellung des Datenschutzes. In Zusammenarbeit mit dem Datenschutzbeauftragten der Universität Duisburg-Essen wird derzeit eine Überprüfung von JACK bezüglich aller relevanten Datenschutzrichtlinien durchgeführt, die voraussichtlich im dritten Projektjahr zu einem positiven Begutachtungsergebnis und der zugehörigen Dokumentation gelangen wird.

A Aufgabenbeispiele

Hier ist die Liste aller Aufgaben zu finden, die für den Mathematik Vorkurs zum Wintersemester 2013/14 entwickelt wurden. In Tabelle [A.1](#) - Tabelle [A.8](#) stehen die Aufgaben, die während des Vorkurses von den Studierenden jederzeit bearbeiten werden konnten, und in Tabelle [A.9](#) sind die Testats Aufgaben zu finden. In der ersten Spalte ist der Name der jeweiligen Aufgabe zu lesen. Danach kann man die Anzahl der eingereichten Lösungen der jeweiligen Aufgabe ablesen. In den nächsten beiden Spalten steht die Anzahl der richtig beantworteten Einreichungen (einmal in absoluten Zahlen und einmal in Prozent). Eine Einreichung wird als richtig bewertet, wenn man ein Minimum von 50 Punkten erreicht hat. In den nächsten Spalten lässt sich stufenweise das Frageformat der jeweiligen Stufe entnehmen.

	Lösungen	richtig	Prozent	Stufe 1	Stufe 2	Stufe 3	Stufe 4	Stufe 5
1 Terme mit einer Variable								
Gesetze	181	140	77,35%	Drop-Down				
Löse die Betragsgleichung	121	55	45,45%	Fill-In	Fill-In	MC		
Löse die Ungleichung	134	93	69,4%	Fill-In	MC	Fill-In	Fill-In	MC
Löse die Ungleichung 1	134	33	24,63%	Fill-In				
Lösungsmenge Gleichung	159	143	89,94%	Fill-In				
Oberfläche eines Würfels	158	120	75,95%	Fill-In	Fill-In	Fill-In		
Term vereinfachen und Gleichung lösen	150	126	84%	Fill-In				
Term vereinfachen und Gleichung lösen 2	261	154	59%	Fill-In	Fill-In	Fill-In	Fill-In	Fill-In
Terme	350	229	65,43%	Fill-In	Fill-In			
Terme aufstellen	167	133	79,64%	Fill-In				
Terme aufstellen und vereinfachen	134	106	79,1%	Fill-In	Fill-In	Fill-In	Fill-In	Fill-In
Terme vereinfachen	189	160	84,66%	Fill-In				
Terme zusammenfassen	164	114	69,51%	Fill-In				
Termumformung	172	156	90,7%	Fill-In				
Termumformungen mit Fehlern	137	109	79,56%	MC				
Textaufgabe Terme	132	120	90,91%	Fill-In	Fill-In			
Ungleichung aufstellen und lösen	134	131	97,76%	Drop-Down	Drop-Down			
Waagschale	136	131	96,32%	Fill-In	Fill-In			
Wasserhöhe im Swimmingpool	131	90	68,7%	Fill-In	Fill-In	Fill-In		
Zahlenrätsel	152	140	92,1%	Fill-In	Fill-In			
2 Terme mit mehreren Variablen								
Binomische Formeln a	112	102	91,07%	MC				
Binomische Formeln b	118	96	81,36%	MC				
Binomische Formeln c	113	84	74,34%	MC				

Tabelle A.1: Alle Aufgaben im Vorkurs, Teil 1.

	Lösungen	richtig	Prozent	Stufe 1	Stufe 2	Stufe 3	Stufe 4	Stufe 5	Stufe 6
Binomische Formeln Ergänzen 1	110	90	81,82%	Fill-In	Fill-In	Fill-In			
Binomische Formeln Ergänzen 2	102	82	80,39%	Fill-In	Fill-In	Fill-In	Fill-In		
Distributivgesetz	114	103	90,35%	MC					
Einsetzungsverfahren	87	55	63,22%	Fill-In	Fill-In	Fill-In			
Faktoriere	111	86	77,48%	Fill-In					
Fläche Term	116	68	58,62%	Fill-In	Fill-In	Fill-In	Fill-In	Fill-In	
Grundstück	114	28	24,56%	Fill-In					
Händeschütteln	96	41	42,71%	Fill-In					
Quadratisches Ergänzen, Binomische Formel	137	75	54,74%	Fill-In	Fill-In	Fill-In			
Term ausklammern	197	39	19,8%	Fill-In					
Terme vereinfachen 2	147	142	96,6%	Fill-In	Fill-In				
Textaufgabe Jahre	99	64	64,65%	Fill-In	Fill-In				
Vereinfache 2	112	90	80,36%	Fill-In					
Wieviel kostet das Auto?	111	79	71,17%	Fill-In	Fill-In	Fill-In	Fill-In		
3 Lineare Funktionen									
Betragsfunktion	86	41	47,67%	Fill-In	Fill-In	Fill-In			
Der Schnittpunkt von zwei Geraden	82	74	88,1%	Fill-In					
Die Gerade und alle ihre Eigenschaften	106	84	79,25%	Fill-In	MC	Fill-In	Fill-In		
Drei Punkte auf einer Geraden	89	68	76,4%	Fill-In	Fill-In	Fill-In			
Eigenschaften von Geraden	95	79	83,16%	Fill-In					
Funktion	100	94	94%	Fill-In	Fill-In	Fill-In	Fill-In		
Funktion bestimmen	81	60	74,07%	Fill-In	Fill-In	Fill-In			
Funktion bestimmen 2	79	73	92,41%	Fill-In	Fill-In				
Funktionsgleichung	84	63	75%	Fill-In					
Getränkckeiste	90	79	87,78%	Fill-In	Fill-In	Fill-In			
Lineare Gleichung	79	57	72,15%	Fill-In					
Nullstelle lineare Gleichung	80	79	98,75%	Fill-In					

Tabelle A.2: Alle Aufgaben im Vorkurs, Teil 2.

	Lösungen	richtig	Prozent	Stufe 1	Stufe 2	Stufe 3	Stufe 4	Stufe 5	Stufe 6	Stufe 7
Punkt bestimmen	79	67	84,81%	Fill-In						
Punkte auf Graph	112	101	90,18%	MC						
4 Systeme linearer Gleichungen										
Additionsverfahren	52	35	67,31%	Fill-In	Fill-In	Fill-In	Fill-In	Fill-In		
Einsetzungsverfahren 1	65	33	50,77%	Fill-In	Fill-In	Fill-In	Fill-In	Fill-In	Fill-In	
Gleichschenkliges Dreieck	80	60	75%	Fill-In	Fill-In	Fill-In	Fill-In			
Gleichsetzungsverfahren	60	23	38,33%	Fill-In	Fill-In	Fill-In	Fill-In	Fill-In		
Graph lineares Gleichungssystem	103	74	71,84%	Fill-In	Fill-In	MC				
Hotel Paluno	62	43	69,35%	Fill-In						
Hühner und Schafe	70	66	94,29%	Fill-In	Fill-In					
Hühner und Schafe 1	87	80	91,95%	Fill-In	Fill-In					
Lineares Gleichungssystem lösen	90	70	77,78%	Fill-In						
Lineares Gleichungssystem lösen mit Anleitung	55	44	80%	Fill-In	Fill-In	Fill-In	Fill-In			
Spätaufgabe: Bestimme das Alter	54	23	42,59%	Fill-In						
Vater und Sohn	54	37	68,52%	Fill-In						
Vater und Sohn 2	91	41	45,05%	Fill-In	Fill-In	Fill-In				
Zugehörigkeit Punkte lineare Graphen	117	81	69,23%	Drop-Down						
Äpfel und Birnen	75	35	46,67%	Fill-In	Fill-In	Fill-In	Fill-In	Fill-In	Fill-In	Fill-In
5 Bruchterme										
Bruch erweitern	92	60	65,22%	Fill-In						
Bruch kürzen	98	80	81,63%	Fill-In	Fill-In					
Bruchgleichung	64	28	43,75%	Fill-In	Fill-In	Fill-In	Fill-In			
Bruchterm Rechteck	85	69	81,18%	MC	Fill-In					
Bruchgleichung	63	32	50,79%	MC	Fill-In	Fill-In	Fill-In	Fill-In		

Tabelle A.3: Alle Aufgaben in Vorkurs, Teil 3.

	Lösungen	richtig	Prozent	Stufe 1	Stufe 2	Stufe 3	Stufe 4	Stufe 5	Stufe 6
Definitionsbereich	84	69	82,14%	MC					
Prozentrechnung und Brüche	94	86	91,49%	MC					
Vereinfachen von Bruchtermen I	62	19	30,65%	Fill-In					
Vereinfachen von Bruchtermen II	70	31	44,29%	Fill-In					
Vereinfachen von Bruchtermen III	87	26	29,89%	Fill-In					
Zahlenrätsel Bruchterme	101	48	47,52%	Fill-In	Fill-In				
6 Quadratwurzel									
Berechne die Wurzel	45	30	66,67%	Fill-In					
Bruch vereinfachen	69	21	30,43%	Fill-In					
Kleiner Kreis, großer Kreis	69	50	72,46%	Fill-In	Fill-In				
Quadratzahl	88	82	93,18%	Fill-In					
Vereinfache Wurzelterm	81	32	39,51%	Fill-In					
Wurzel ziehen	86	78	90,70%	Fill-In					
Wurzelgleichung	73	42	57,53%	Fill-In	Fill-In	Fill-In	Fill-In	Fill-In	Fill-In
Wurzeln vereinfachen	66	33	50%	Fill-In					
Wurzeln vergleichen	61	51	83,61%	MC					
Wurzelterm	73	47	64,38%	Fill-In					
Wurzelterm div	80	59	73,75%	Fill-In					
Wurzelterm plus	82	16	19,51%	Fill-In					
7 Quadratische Funktion und ihr Graph									
allgemeine quadrat. Funktion	61	33	54,10%	Fill-In					

Tabelle A.4: Alle Aufgaben im Vorkurs, Teil 4.

	Lösungen	richtig	Prozent	Stufe 1	Stufe 2	Stufe 3	Stufe 4	Stufe 5
Finde die Nullstellen der Parabel	51	41	80,39%	Fill-In				
Finde die Parabelgleichung	47	27	57,45%	Fill-In				
Funktionsvorschrift	56	32	57,14%	Fill-In				
Kurvendiskussion einer Parabel	50	40	80%	Drop-Down	MC	Fill-In	Fill-In	Fill-In
Normalparabel	53	31	58,49%	Fill-In				
Scheitelpunkt	82	55	67,07%	Fill-In				
Scheitelpunktsform	63	43	68,25%	Fill-In				
Verlauf eines Graphen	48	25	52,08%	Drop-Down				
Zwei Parabeln	40	16	40%	Fill-In	Fill-In	Fill-In		
8 Quadratische Gleichungen								
Berechne die Zahl	33	31	93,94%	Fill-In	Fill-In			
Bestimme die fehlenden Parameter	27	9	33,33%	Fill-In	Drop-Down	Drop-Down		
Bestimme die Seitenlänge eines Rechtecks anhand von Diagonalen und Umfang	40	15	37,5%	Fill-In	Fill-In	Fill-In		
Der goldene Schnitt	24	4	16,67%	Fill-In	Fill-In	Fill-In		
Der Satz von Vieta	47	40	85,11%	Fill-In	Fill-In			
Der Scheitelpunkt einer Parabel	31	12	38,71%	Fill-In				
Die Oberfläche des Quaders	43	19	44,19%	Fill-In	Fill-In	Fill-In	Fill-In	
Finde die Lösung der Bruchgleichung	41	37	90,24%	Fill-In	Fill-In			
Linearfaktoren	48	30	62,50%	Fill-In				
Lösen quadratischer Gleichungen	30	14	46,67%	Fill-In				
Lösungsmenge einer Bruchgleichung	45	34	75,56%	Fill-In				

Tabelle A.5: Alle Aufgaben im Vorkurs, Teil 5.

	Lösungen	richtig	Prozent	Stufe 1	Stufe 2	Stufe 3	Stufe 4
quadratische Gleichung lösen	25	14	56%	Fill-In			
Rechteck	30	22	73,33%	Fill-In	Fill-In	Fill-In	
Textaufgabe Quadratische Gleichungen	33	28	84,85%	Fill-In	Fill-In	MC	Fill-In
Ungleichung lösen	28	23	82,14%	Fill-In	Drop-Down		
9 Flächensätze am Dreieck							
Der Flächeninhalt eines Dreiecks	37	34	91,89%	MC			
Die Raumdiagonale des Würfels	38	28	73,68%	Fill-In	Fill-In		
Drei Kreise	28	2	7,14%	Fill-In			
Dreieck	38	36	94,74%	Fill-In			
Flächeninhalt Dreieck	40	25	62,50%	Fill-In			
Großes Dreieck, kleines Dreieck	28	22	78,57%	Fill-In	Fill-In		
Höhe eines Dreiecks	31	30	96,77%	Fill-In	Fill-In		
Höhensatz	30	25	83,33%	Fill-In			
Katheten eines rechtwinkligen Dreiecks	39	17	43,59%	Fill-In			
Kreistangenten	31	30	96,77%	MC	Fill-In		
Maibaum	33	29	87,88%	Fill-In			
Quadrat	53	35	66,04%	Fill-In	Fill-In		
Seitenlänge Dreieck	46	11	23,91%	Fill-In			
Stumpfer, spitzer oder rechter Winkel?	36	27	75%	Drop-Down	Drop-Down	Drop-Down	Drop-Down
Wie ist der Flächeninhalt des Rechtecks?	27	9	33,33%	Fill-In			
10 Potenzen							
Alternierendes Produkt	20	15	75%	MC			
Berechne die Potenzen	38	34	89,47%	Fill-In	Fill-In	Fill-In	Fill-In
Besondere Produkte	29	16	55,17%	Fill-In	Fill-In	Fill-In	

Tabelle A.6: Alle Aufgaben im Vorkurs, Teil 6.

	Lösungen	richtig	Prozent	Stufe 1	Stufe 2	Stufe 3	Stufe 4
Bevölkerungsentwicklung	25	9	36%	Fill-In			
Forme die Potenzen um	34	23	67,65%	Fill-In	Fill-In	Fill-In	Fill-In
Glücksspiel	26	14	53,85%	Fill-In			
Lebensdauer eines Elementarteilchens	24	8	33,33%	Fill-In			
Lösungsmenge Potenzen	31	26	83,87%	Fill-In			
Potenzgesetze	35	31	88,57%	MC			
Potenzgesetze 2	37	35	94,59%	MC			
Potenzgesetze 3	37	35	94,59%	Fill-In			
Schreibe als eine Potenz	31	31	100%	Fill-In	Fill-In	Fill-In	Fill-In
Vereinfache	27	18	66,67%	Fill-In	Fill-In	Fill-In	
Vereinfache mit Potenzgesetzen	36	24	66,67%	Fill-In			
Vereinfache Potenz	36	24	66,67%	Fill-In			
Vereinfache Potenzen	28	12	42,86%	Fill-In			
Wahr oder falsch?	27	8	29,63%	MC			
Wie hoch springt der Ball?	40	0	0%	Fill-In	Fill-In		
Wurzel gezogen	36	23	63,89%	Fill-In			
Würfel formen	31	28	90,32%	Fill-In	Fill-In	Fill-In	
11 Exponentialfunktionen							
Bestimme das Produkt	35	14	40%	Fill-In			
Bevölkerung Italiens	19	5	26,32%	Fill-In	Fill-In	Fill-In	
Eigenschaften der Exponentialfkt.	29	19	65,52%	MC			
Finde die Exponentialfkt.	23	17	73,91%	Fill-In			
Finde die Exponentialfkt. 2	19	15	78,95%	Fill-In	Fill-In		
Logarithmen	34	17	50%	Fill-In			
Logarithmus Regeln	26	10	38,46%	Fill-In			
Lösungsmenge Exponentialfkt.	19	13	68,42%	Fill-In			
Lösungsmenge Logarithmusfkt.	19	6	31,58%	Fill-In			
Vereinfache Logarithmus	21	10	47,62%	Fill-In			
Zinsen Guthaben	32	18	56,25%	Fill-In	Fill-In	Fill-In	

Tabelle A.7: Alle Aufgaben im Vorkurs, Teil 7.

	Lösungen	richtig	Prozent	Stufe 1	Stufe 2	Stufe 3	Stufe 4
12 Trigonometrie							
Additionssätze	14	1	7,14%	Fill-In			
Beziehung zwischen den trigonometrischen Funktionen	15	8	53,33%	Fill-In			
Cosinus	22	19	86,36%	MC	Fill-In	Fill-In	
Gleichung beweisen	14	4	28,57%	Fill-In	Fill-In	Fill-In	
Sinus	18	15	83,33%	MC	Fill-In	Fill-In	
trigonometrische Funktion	15	14	83,33%	Fill-In			
Trigonometrische Funktionen am Einheitskreis	15	9	60%	MC			
Vereinfache trigonometrischen Term	13	5	38,46%	Fill-In			
Wichtige Werte von Sinus und Kosinus	15	12	80%	Fill-In	Fill-In	Fill-In	Fill-In
Winkel Berechnung	15	12	80%	Fill-In			

Tabelle A.8: Alle Aufgaben im Vorkurs, Teil 8.

Aufgabe	Frageformat	Woche
Berechne die Wurzel	Fill-In	2
Bestimme das Produkt	Fill-In	3
Bestimme die fehlende Länge	Fill-In	3
Bestimme die Funktionsgleichungen	Fill-In	1
Bestimme die Steigung der Geraden	Fill-In	1
Der richtige Exponent	Fill-In	3
Der Umfang eines Rechtecks	Fill-In	1
Die Diagonalen des Parallelogramms	Fill-In	3
Dreieck im Quadrat	Fill-In	3
Ein bisschen Trigonometrie	Fill-In	3
Einschränkende Bedingungen bei Wurzeltermen	MC	2
Finde den Fehler	MC	1
Flächeninhalt Kreisring	Fill-In	3
Korrekte Umformung gesucht	MC	1
Logarithmusfunktion	Fill-In	3
Löse das lineare Gleichungssystem	Fill-In	1
Nullstellen genügen Gleichung	Fill-In	2
Nullstellen gesucht	MC	2
Nüsse	MC	1
Potenzgleichung lösen	Fill-In	3
Problemlösung gesucht	Fill-In	1
Quadratische Funktion	Fill-In	2
Quadratische Gleichung	MC	2
Quadratische Ungleichung	Drop-Down	2
Rechteck im Dreieck	Fill-In	2
Ungleichung mit zwei Variablen lösen	Fill-In	2
Welchen Wert kann die Summe annehmen	MC	1
Wie lang ist der Einkaufswagen?	Fill-In	1
Wie lautet die quadratische Funktion	Fill-In	2
Zahlenrätsel	Fill-In	3

Tabelle A.9: Alle Aufgaben aus den Testaten und der Abschlussklausur im Vorkurs.

Literaturverzeichnis

- [FGKK⁺13] FISCHOTTER, Melanie ; GOEDICKE, Michael ; KURT-KARAOGLU, Filiz ; SCHWINNING, Nils ; STRIEWE, Michael: Erster Jahresbericht zum Projekt "Bildungsgerechtigkeit im Fokus"(Teilprojekt 1.2 - "Blended Learning") an der Fakultät für Wirtschaftswissenschaften / ICB, Universität Duisburg-Essen. 2013. – Forschungsbericht
- [SKKS⁺13] SCHYPULA, Melanie ; KURT-KARAOGLU, Filiz ; SCHWINNING, Nils ; STRIEWE, Michael ; GOEDICKE, Michael: Beobachtungen zur Motivation der Studierenden bei verschiedenen Frageformaten. In: ANDREAS BREITER, Christoph R. (Hrsg.): *DeLFI 2013: Die 11. e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V. (GI), 8. - 11 September 2013, Bremen* Bd. 218, GI, 2013 (LNI). – ISBN 3-88579-612-1, S. 35-45

Previously Published ICB Research Reports

2014

No 58 (March 2014)

Breitschwerdt, Rüdiger; Heß, Michael: »Konzeption eines Bezugsrahmens zur Analyse und Entwicklung von Geschäftsmodellen mobiler Gesundheitsdienstleistungen – Langfassung«

No 57 (March 2014)

Heß, Michael; Schlieter, Hannes (Hrsg.): »Modellierung im Gesundheitswesen – Tagungsband des Workshops im Rahmen der "Modellierung 2014"«

2013

No 56 (July 2013)

Svensson, Richard Berntsson; Berry, Daniel M.; Daneva, Maya; Doerr, Joerg; Espana, Sergio; Herrmann, Andrea; Herzwurm, Georg; Hoffmann, Anne; Pena, Raul Mazo; Opdahl, Andreas L.; Pastor, Oscar; Pietsch, Wolfram; Salinesi, Camille; Schneider, Kurt; Seyff, Norbert; van de Weerd, Inge; Wieringa, Roel; Wnuk, Krzysztof (Eds.): »19th International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ 2013). Proceedings of the REFSQ 2013 Workshops CreaRE, IWSPM, and RePriCo, the REFSQ 2013 Empirical Track (Empirical Live Experiment and Empirical Research Fair), the REFSQ 2013 Doctoral Symposium, and the REFSQ 2013 Poster Session"«

No 55 (May 2013)

Daun, Marian; Focke, Markus; Holtmann, Jörg; Tenbergen, Bastian »Goal-Scenario-Oriented Requirements Engineering for Functional Decomposition with Bidirectional Transformation to Controlled Natural Language. Case Study "Body Control Module"«

No 54 (March 2013)

Fischotter, Melanie; Goedicke, Michael; Kurt-Karaoglu, Filiz; Schwinning, Nils; Striewe, Michael »Erster Jahresbericht zum Projekt "Bildungsgerechtigkeit im Fokus" (Teilprojekt 1.2 – "Blended Learning") an der Fakultät für Wirtschaftswissenschaften«

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: »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«

2009

No 36 (December 2009)

Strecker, Stefan: »Ein Kommentar zur Diskussion um Begriff und Verstandnis 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; Bruno Muller–Clostermann: »CapSys: A Tool for Macroscopic Capacity Planning«

No 25 (August 2008)

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

No 24 (August 2008)

Frank, Ulrich: »The MEMO Meta Modelling Language (MML) and Language Architecture - 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 International Workshop on Variability Modelling of Software–intensive Systems.«

2007

No 21 (September 2007)

Eicker, Stefan; Annett Nagel; Peter M. Schuler: »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 Wirtschaftsinformatik: 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«

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. Echte 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. Ing. E. Rathgeb Computer Network Technology	Computer Network Technology
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