

Masterarbeit

**Adaptives Informationssystem  
für  
Mathematische Lernanwendungen**

Michael Steyrer

---

Institut für Informationssysteme und Computer Medien  
Technische Universität Graz



Betreuer: Univ.-Doz. Dipl.-Ing. Dr. techn. Martin Ebner

Graz, im August 2012

## **Kurzfassung**

Den entscheidenden Grundstein für die Ausbildung von individuellen Lernkompetenzen bildet der Unterricht in Volksschulen, welcher Kinder in besonderem Maße prägt. Ausschlaggebend für erfolgreiche Lernfortschritte sind fachgemäße Lehrmethodiken und eine effektive Art der Informationsvermittlung durch Lehrkräfte. Eine immer größere Rolle in der Lernpädagogik nehmen moderne Unterrichtsformen ein, welche durch die Erforschung von Learning Analytics an Bedeutung gewinnen. Ein besseres Verständnis von Lernern zu erhalten, ist Ziel der Lernanalytik. Eng damit verbunden ist die Optimierung von Lehre und Lernprozessen, welche zur Steigerung von individueller Leistungsfähigkeit beitragen. Basis dafür sind innovative Lernsysteme und deren umfassende Analyseverfahren. Durch gezielte Aufzeichnungen von Lernaktivitäten werden geeignete Rückschlüsse an Lehrende geliefert. Entscheidend für den Erfolg solcher Systeme ist die stetige Interaktion mit Lehrpersonen. Die fortschreitende Vernetzung ermöglicht es, leistungsrelevante Datenabgriffe in Lernprozessen zu automatisieren. Richtungsweisende Analyseverfahren wie die Methode der Sozialen Netzwerkanalyse streben die Ermittlung von gezielten Lernvorhersagen an, um Lehrenden bei auftretenden schülerspezifischen Schwierigkeiten geeignete Maßnahmen zur positiven Leistungsbeeinflussung zu liefern. Mathematische Lernsoftware ist beispielgebend für computerbasierte Lernunterstützung für Schülerinnen und Schüler. Wesentliche Eigenschaften von Lernanwendungen umfassen eine intuitive Behandlung und die Automatisierung von an Lernprozessen beteiligten Verfahrensweisen. Nur die gemeinsame Ergänzung und dadurch vereinte Synergien von Lernprogramm und Lehrkörper spiegeln Lernerfolg wider.

Diese Arbeit stellt die Entwicklung des Mathe-Multi-Trainers, einer adaptiven Lernapplikation zur Aneignung der mehrstelligen Multiplikation, in den Vordergrund. Das zweisprachige webbasierte Programm ist in volksschulgerechtem Design ausgeführt und liefert mehrstellige Multiplikationen zur Lösungsfindung durch Schülerinnen und Schüler, welche das Einmaleins beherrschen. Der Trainingsmodus des Programms passt sich in individueller Weise auf die jeweilige Lernkompetenz von Benutzerinnen und Benutzern an. Die Stellenzunahme der Faktoren und die Unterscheidung von Multiplikationen mit und ohne Übertrag bilden die Grundlage einer ausgereiften und universellen Beispielgebung. Die Lernapplikation ist in der Lage, beliebige Rechenfehler zu erkennen und diese für Lehrkräfte mittels einer breitgefächerten Statistikauswertung verfügbar zu machen. Der Mathe-Multi-Trainer steht Lehrenden bei der mathematischen Leistungsbeurteilung von Schülerinnen und Schülern unterstützend zur Seite und verhilft Lernern durch gezieltes Training zu einer erfolgreichen Lösungsfindung von vielschichtigen Multiplikationsrechnungen. Intention des Lerntainers ist der reale Einsatz in Volks- und Mittelschulen.

## **Abstract**

The acquirement of knowledge and learning expertise is primarily based on instruction in elementary schools which is especially important for children. The right way of education as well as competent and promising methods of instruction lead to a desired and successful learning progress. Beneficial learning outcomes depend as well on instructors themselves by means of individual teaching methods. Regarding educational science the research area of learning analytics provides a multitude of new ways of modern teaching. The purpose of this scientific field is to get to know and to understand learners. Another crucial aspect of learning analytics is the optimization of instruction and learning processes for increasing an individual's learning performance. This can be based on innovative learning systems which provide plenty of comprehensive analysis techniques. Instructors are given a large bulk of data of recorded learning activities in order to draw adequate conclusions for learners. The secret to success of these kind of systems lies within its continuous interaction with instructors. The high degree of Internet usage allows automated processes for recordings of performance data relevant to learning. Trend-setting analysis techniques such as Social Network Analysis aim at providing well-directed learning predictions for students. In case of the emergence of individual learning difficulties teachers analyze the generated predictions for helping students to improve in performance.

Computer-based support for students concerning learning is represented in this paper by mathematics educational software. Fundamental characteristics of educational applications include intuitional treatment and automation of learning processes. Synergies of both program and instructor and thus learning success can only be achieved by a combined deployment.

The main part of this thesis deals with the implementation of an adaptive learning application for acquiring knowledge of multi-digit multiplication named Multi-Math-Coach. The web-based program operates in a bilingual environment especially designed for elementary school purposes. It provides multi-digit multiplications for students already mastering multiplication tables. An important part of the application is represented by its training mode which acts adaptively to each student's learning expertise. The increase of digits of multiplication factors as well as the distinction of calculations by its carry mode form the basics of a perfected and universal generation of multiplications. The learning application is able to identify plenty of common errors of calculation which are all visible for teachers through a comprehensive statistical evaluation operation. The Multi-Math-Coach application assists instructors in evaluating students' performances and supports learners to a successful acquisition of sufficient mathematical knowledge for solving multi-digit multiplications correctly. The program's main intention is to be applied in primary and secondary schools in real life.

Deutsche Fassung:  
Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008  
Genehmigung des Senates am 1.12.2008

## EIDESSTÄTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Graz, am .....

.....  
(Unterschrift)

Englische Fassung:

## STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

.....  
date

.....  
(signature)

## **Danksagung**

Zu Beginn möchte ich mich bei Herrn Martin Ebner recht herzlich für die angenehme Betreuung dieser Arbeit bedanken. Ein weiterer Dank gilt Herrn Martin Schön für die vielen Ideen und das brauchbare Feedback bei der Entwicklung des Lerntrainers.

Ganz besonders möchte ich mich bei meinen Eltern für die immense Hilfe und Unterstützung während meiner gesamten Studienzzeit bedanken. Vor allem gilt mein Dank dafür, dass mir die Möglichkeit gegeben wurde, neben meinem Studium ein überdurchschnittliches Maß an außergewöhnlicher Lebenserfahrung zu sammeln.

Ich widme diese Arbeit meinem Neffen Johann.

Graz, im August 2012

Michael Steyrer



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Struktur der Arbeit . . . . .	2
<b>2</b>	<b>Learning Analytics</b>	<b>3</b>
2.1	Learning Management Systems (LMS) . . . . .	6
2.1.1	Plattform für Lernanalytik . . . . .	8
2.1.2	Recommender Systems . . . . .	8
2.2	Vorhersage- und Ursachenmodelle . . . . .	12
2.3	Diskursanalyse . . . . .	12
2.4	Verhaltensbeobachtung . . . . .	14
2.5	Social Network Analysis (SNA) . . . . .	16
2.5.1	Das eTwinning-Netzwerk . . . . .	19
2.6	Data-Mining . . . . .	19
2.7	Logging . . . . .	21
2.7.1	Lecture Capture . . . . .	23
2.8	Kritische Stimmen zu Learning Analytics . . . . .	24
<b>3</b>	<b>Mathematische Lernsoftware</b>	<b>25</b>
3.1	DynaMath . . . . .	26
3.1.1	Erzeugung der Aufgaben . . . . .	28
3.1.2	Bewertungsergebnisse . . . . .	29
3.2	TORUS . . . . .	29
3.3	TutorIT . . . . .	31
<b>4</b>	<b>Technische Umsetzung des Prototyps</b>	<b>39</b>
4.1	Verwendete Technologien . . . . .	39
4.1.1	PHP 5 . . . . .	39
4.1.2	MySQL . . . . .	40
4.1.3	CSS . . . . .	40
4.1.4	jQuery . . . . .	41
4.1.5	Zend Framework . . . . .	41
4.2	Konzept und Funktionalität . . . . .	42
4.2.1	Allgemeiner Aufbau der Implementation . . . . .	44
4.3	Layout der Benutzeroberfläche . . . . .	46
4.4	Generierung der Beispiele . . . . .	49
4.4.1	Entwicklung des Algorithmus . . . . .	50
4.5	Evaluierung der Eingaben . . . . .	58
4.5.1	Auswertung von Multiplikationen ohne Übertrag . . . . .	60

4.5.2	Auswertung von Multiplikationen mit Übertrag . . . . .	63
4.6	Datenbankdesign . . . . .	67
4.6.1	Tabelle multipliers . . . . .	67
4.6.2	Tabelle answers . . . . .	67
4.6.3	Tabelle basicAnswers . . . . .	70
4.6.4	Tabelle basicCarryAnswers . . . . .	70
4.6.5	Tabelle studentExpertise . . . . .	71
4.7	Statistikauswertung . . . . .	72
4.8	Programmablauf im Trainingsmodus . . . . .	75
4.9	Bereitstellung von Webservices . . . . .	80
<b>5</b>	<b>Schlussfolgerung und Ausblick</b>	<b>81</b>
	<b>Literaturverzeichnis</b>	<b>83</b>



# Abbildungsverzeichnis

2.1	Pyramide zur Verwendung von Daten an der University of Phoenix [Sharkey 2011]. . . . .	4
2.2	Eigenschaften einer Universität im Laufe der Zeit, nach [Shum und Ferguson 2011] . . . . .	5
2.3	Dreistufiges Modell der Teaching Analytics nach [Vatrapu et al. 2011]. . . . .	7
2.4	Individueller Performance-Report für Studenten, [Bader-Natal und Lotze 2011]. . . . .	9
2.5	Mögliche Verbindungsarten in einer Konversation [De Liddo et al. 2011]. . . . .	13
2.6	Ausschnitt der Darstellung eines Netzwerkdiagramms eines Diskussionsforums durch SNAPP [Merhbi 2011]. . . . .	18
2.7	Beispiel zur Auswertung einer Log-Datei [Blikstein 2011]. . . . .	23
3.1	Das Auswahlwerkzeug des AuthorIT Systems [Scandura 2011]. . . . .	34
3.2	TutorIT im Schülermodus mit einer Multiplikationsaufgabe [Scandura 2011]. . . . .	35
3.3	Ausführung von TutorIT im Tutormodus [Scandura 2011]. . . . .	37
4.1	Die Eingangsseite des Mathe-Multi-Trainers. . . . .	46
4.2	Die Lernanwendung während der Beantwortung einer Multiplikationsaufgabe. . . . .	47
4.3	Die korrekte Lösung einer Multiplikationsaufgabe ohne Übertrag der Kompetenz II und Gruppe 6. . . . .	48
4.4	Der Mathe-Multi-Trainer nach fehlerhafter Lösung einer Multiplikationsaufgabe mit Übertrag der schwierigsten Kompetenzkategorie und Gruppe. . . . .	48
4.5	Beispiel für die Statistikauswertung der Lernapplikation nach Selektion eines bestimmten Users. . . . .	49
4.6	Zwei Eingabefehler (ERR_INPUT) beim Lösen einer Multiplikationsaufgabe. . . . .	62
4.7	Demonstration von 1x1 Fehlern (ERR_1x1) und einer korrekten Addition mit Folgefehlern (OK_END_ERR) beim Lösen eine übertragslosen Multiplikation. . . . .	63
4.8	Demonstration von Fehlern mit vertauschter Eintragung (ERR_REV_IN) und dem Additionsfehler ERR_END_OK beim Lösen einer übertragslosen Multiplikation. . . . .	64
4.9	Demonstration der Zustände OK_MENTAL und OK_NOCARRY sowie der Fehler ERR_UNKNOWN und ERR_END_ADD beim Lösen einer Multiplikation mit Übertrag. . . . .	65
4.10	Demonstration der Zustände OK_ERR, OK_END_ERR, ERR_REV_IN, ERR_1x1 und ERR_ADD beim Lösen einer Multiplikation mit Übertrag . . . . .	66
4.11	Demonstration der Zustände OK_NOCARRY und OK_END_ERR sowie des Fehlers ERR_NOADD beim Lösen einer Multiplikation mit Übertrag. . . . .	66
4.12	Das Datenbankmodell des Mathe-Multi-Trainers. . . . .	68
4.13	Beispielansicht der Übersichtsseite der Statistikauswertung des Mathe-Multi-Trainers mit zwei registrierten Klassen. . . . .	72
4.14	Die Detailstatistik für den Benutzer <i>Max Musterschüler</i> bei Auswahl aller möglichen Untertabellen für eine bestimmte Aufgabenstellung. . . . .	74

4.15	Ausschnitt aus der Detailaufstellung für eine übertragsbehaftete Multiplikation. . . .	75
4.16	Programmablaufplan des Trainingsmodus des Mathe-Multi-Trainers. . . . .	78
4.17	Das Kompetenzmodul des vorgestellten Trainingsalgorithmus. . . . .	79

# Tabellenverzeichnis

2.1	Abstraktionsebenen analytischer Darstellungsformen nach [Suthers und Rosen 2011].	21
3.1	Die vier Kategorien möglicher Multiplikationen mit steigendem Komplexitätsgrad [Gerber et al. 1994]. . . . .	27
4.1	Die drei Kompetenzstufen mit den acht Problemgruppen und den entsprechenden Multiplikationsmustern des Mathe-Multi-Trainers. . . . .	43
4.2	Ziffernwerte mit entsprechenden Auftrittswahrscheinlichkeiten für einen Multiplikator der Problemgruppe 1. . . . .	51
4.3	Erster Testversuch für die Ermittlung eines Multiplikanden der Problemgruppe 1. . .	52
4.4	Ergebnisse von 100 Testläufen zur Ermittlung von Auftrittswahrscheinlichkeiten ( $p$ ) für die Einerstelle eines Multiplikanden der Problemgruppe 1. . . . .	53
4.5	Finale Einstellungen für die Ermittlung eines Multiplikanden der Problemgruppe 1. .	53
4.6	Vorschrift für die Auftrittswahrscheinlichkeiten eines Multiplikators aus Gruppe 2. .	54
4.7	Zuweisungsregeln für die Generierung von Multiplikanden für Problemgruppe 2 und 3.	54
4.8	Auftrittswahrscheinlichkeiten für Ziffern eines Multiplikators der Problemgruppe 3. .	54
4.9	Alle möglichen Multiplikatoren und deren Auftrittswahrscheinlichkeiten für die Problemgruppen 3 bis 6 (Kompetenzkategorie II). . . . .	55
4.10	Alle möglichen Multiplikatoren einschließlich der Auftrittswahrscheinlichkeiten für die Problemgruppen 7 und 8 (Kompetenzkategorie III). . . . .	56
4.11	Die möglichen Ziffern inklusive Auftrittswahrscheinlichkeiten von Multiplikatoren für übertragsbehaftete Beispiele. . . . .	57
4.12	Generiervorschrift für Multiplikanden bei Aufgabenstellungen mit Übertrag. . . . .	58
4.13	Alle feststellbaren OK-Zustände des Mathe-Multi-Trainers und deren Vorkommen. . .	61
4.14	Alle feststellbaren Fehler des Mathe-Multi-Trainers und deren Vorkommen. . . . .	61
4.15	Alle möglichen Werte des Attributs <code>productCategory</code> der Antworttabelle einschließlich entsprechender Erklärungen. . . . .	69



# 1 Einleitung

Der Mathematikunterricht in Volksschulen ist für viele Schülerinnen und Schüler der erste Schritt in eine komplexe Zahlenwelt, auf welcher ein Großteil aller Wissenschaften beruht. Die Mathematik lässt Menschen zu einem gewissen Grade also ein Leben lang nicht mehr los. Umso wichtiger erscheint es, das A und O dieser *Wissenschaft der Zahlen* zu beherrschen. Die Voraussetzung dafür stellt die Vermittlung der mathematischen Grundkenntnisse in Volks- und Mittelschulen dar. Einen wesentlichen Baustein nimmt dabei die Rolle der Lehrperson ein, welche für den Wissensaufbau bei Schülern verantwortlich ist. Aufgrund des geringen Alters der Schüler ist die Art der Vermittlung ein sehr sensibler Prozess. Eine mangelhafte Lehrfähigkeit stellt für Kinder von Anfang an ein unüberbrückbares Hindernis beim Aneignen von Basiswissen dar und führt zur Entstehung von Unsicherheiten bei der Behandlung von mathematischen Aufgabenstellungen, welches sie im Laufe der Jahre für mathematische Problemstellungen meist nicht mehr loslässt. Daher wird es für Lehrpersonen als unerlässlich angesehen, so früh als möglich geeignete Methoden zur Leistungssteigerung von Lernern und zur Verbesserung der Wissenslehre zu kennen und anzuwenden.

Ein wichtiges Forschungsgebiet stellt in dieser Hinsicht die Analyse von lernspezifischen Daten dar, welche besagte Optimierung von Lehre und Lernleistung zum Ziel hat. In der vorliegenden Arbeit werden aktuelle Untersuchungen zu *Learning Analytics* ausgearbeitet. Schwerpunkte bilden Lernsysteme und deren spezielle Analyseverfahren, welche ein schnelleres Erkennen und damit Eingreifen bei Lernproblemen von Schülern fördern. Je früher Lernschwierigkeiten erkannt werden, desto besser kann Problemen entgegengewirkt werden und desto schneller werden schülerspezifische Unsicherheiten für bestimmte Aufgabengebiete zurückgehen.

Eine immer größere Rolle bei der Unterstützung von Lehrkräften nimmt computergestützte Lernsoftware ein. In dieser Arbeit wird das Hauptaugenmerk auf mathematikbasierte Lernanwendungen gelegt. Die Eigenschaften und Funktionalitäten derartiger Programme variieren stark und sind nicht immer für den Unterricht geeignet. Besonders die sensiblen Bereiche Volks- und Mittelschule setzen für Lernprogramme voraus, nicht das eigentliche Lernziel zu behindern oder gar zu gefährden oder, im ungünstigsten Fall, einer Lehrperson in deren Unterrichtsmethoden entgegenzuwirken. Die positiven Charakteristiken bereits bestehender Lernanwendungen sollen schließlich zum wesentlichen Ziel dieser Arbeit führen – der Implementierung einer mathematischen Lernapplikation zur Erlernung der mehrstelligen Multiplikation. Im Zuge der E-Learning-Initiative an der Technischen Universität Graz wurden bereits erste Lernanwendungen für Volksschüler entwickelt und getestet, darunter ein 1x1-Trainer<sup>1</sup>. Das Lernprogramm für mehrstellige Multiplikationen soll daran anknüpfen und als Zielgruppe jene Schülerinnen und Schüler führen, welche das Einmaleins bereits beherrschen. Wichtige Eigenschaft des Lerntainers soll die bestmögliche Unterstützung von Lehrkräften bei der Leistungsermittlung ihrer Schüler während des Lernprozesses des mathematischen Fachbereichs der mehrstelligen Multiplikation sein. Das Programm soll Fehler von individuellen Lösungsfindun-

---

<sup>1</sup> <http://mathe.tugraz.at/index/play> [Zugriff am 21.08.2012]

gen erkennen können und diese an die zuständigen Lehrpersonen weiterleiten. Ein grundlegendes Merkmal soll darüber hinaus die Adaptivität darstellen. Das bedeutet, die Lernapplikation soll sich bei dessen Verwendung durch Volks- und Mittelschüler adaptiv an deren jeweilige Leistungsstärken anpassen. Diese Charakteristik stellt die Lehrfähigkeit des Programmes her und soll im weiteren Sinne dazu führen, dass durch die Interaktion mit der Lernanwendung eine Erweiterung und Stärkung der Kompetenzen von Schülern stattfindet.

## 1.1 Struktur der Arbeit

Der wissenschaftliche Schwerpunkt der vorliegenden Arbeit beruht im nachfolgenden Kapitel 2 auf der Behandlung von Merkmalen und Methoden von *Learning Analytics*. Es werden grundlegende allgemeine Lernsysteme und Plattformen vorgestellt, und darauf basierende Analysemodelle untersucht und erläutert.

Abschnitt 3 handelt von mathematischer Lernsoftware. Dabei werden zuerst zwei ältere Anwendungen und deren Funktionsweise gezeigt, ehe ein zeitgemäßes Lernsystem namens *TutorIT* vorgestellt wird. Dieses unterstützt eine Vielzahl von mathematischen Fachbereichen und stellt zudem ein eigenes Konfigurationstool bereit.

Der Prototyp der entwickelten Lernapplikation, der *Mathe-Multi-Trainer* wird schließlich in Kapitel 4 präsentiert. Es werden alle Funktionalitäten und Eigenschaften der Lernapplikation in detaillierter Weise dargestellt.

Es sei angemerkt, dass in vorliegender Arbeit personenbezogene Ausdrücke wie Schüler, Lehrer, Anwender, Benutzer und weitere hier nicht angeführte Bezeichner durchgehend geschlechtsneutral verwendet werden. Das heißt, dass die Bedeutung dieser Wörter sowohl weibliche als auch männliche Personen fortwährend einschließt, sofern dies nicht extra hervorgehoben wird.

## 2 Learning Analytics

Der immense technologische Fortschritt und die damit einhergehende Forderung nach neuen und zukunftsweisenden Lehrmethoden stellt den Bereich der Learning Analytics in den Blickpunkt der Entwicklung ebensolcher moderner Unterrichtsmethoden. Bereits vor über 20 Jahren – um das Jahr 1990 – setzte eine Diskussion über die Qualitätsverbesserung der Lehre mittels Computer auf Universitäten ein. Eine Reihe von innovativen Programmen zur Erschließung neuer leistungsfähiger Lehrmethoden und neuem Wissen wurde im Laufe der Jahre gefördert und entwickelt. Der rasante Aufstieg der Internettechnologie brachte einen steigenden Einsatz von neuartigen vernetzten Kommunikationsformen im Bereich des Lernens mit sich. Der Prozess von Lehre und Lernen hat im Laufe der Jahre enorm an Komplexität zugenommen. Der aktuelle Fokus der Lernforschung liegt auf Methoden der Lernanalytik. [Lockyer und Dawson 2011]

Lernanalytik verfolgt nach [Bader-Natal und Lotze 2011] ein besseres Verständnis von Lernen. Der Theoretiker George Siemens versteht unter Lernanalytik gemäß [Siemens 2010b] die Auffindung von Information, um Lernvorhersagen zu treffen. Die Untersuchungsmethoden arbeiten mit lernbasierten Daten und verschiedenen Analysemodellen. Dabei beschäftigt sich die Lernanalytik nicht nur mit Analyseprozessen. Beruhend auf diesen Untersuchungen umfasst sie viele andere Bereiche wie das Eingreifen bei negativen Lernerfolgen, eine Prognose ebendieser und das Anpassen von persönlichen Lernplänen. Die Prognose ist ausschlaggebend für Anpassung, Personalisierung und Eingriff in den Lernprozess und folgt immer den Resultaten der Analysephase.

Lernanalytik wird manchmal auch als Akademische Analytik bezeichnet. Sie soll für die Untersuchung und Verbesserung der Leistung des Lerners stehen und darüberhinaus das Auftreten von Lernerfolgen kennzeichnen [Fournier et al. 2011]. [Pardo und Kloos 2011] zufolge umfasst Lernanalytik Bereiche der Business Intelligence, Geschäftslogik, des Data-Mining und der Ablaufanalyse. Dabei wird die Lernverbesserung durch sogenannte Aktuatoren erzielt. Sie entstehen durch Analyse und Auswertung von zuvor erfassten Daten. Die Autoren unterteilen den Prozess der Lernanalytik in fünf Schritte. Neben der Datenerfassung und der individuellen Auswertung für Benutzer sind dies das Treffen von Lernvorhersagen, in weiterer Folge das Setzen von Lerneingriffen und, schlussendlich, die stetige Verbesserung dieses Ablaufes.

Die „1st International Conference on Learning Analytics and Knowledge 2011“ beschreibt Lernanalytik als Auswertung von Daten über Lerner und deren Kontext [Siemens 2010a]. Ziel bleibt die Lernoptimierung, wobei auch die Lernumgebung dahingehend verbessert werden soll. Zur Lernanalyse sollen sowohl quantitative als auch qualitative Daten herangezogen werden [Fournier et al. 2011].

Lernanalytik ist eng verbunden mit Lerndesign – entworfene Lernmuster verkörpern dabei Lehr- und Lernmethoden. Sie sind üblicherweise flexibel in variablen Bereichen des Lernens einsetzbar. Während sich die Analytik hauptsächlich um Lerner dreht, zielt das Design mehr auf Lehrende ab. Dieser ist für den Entwurf von Methoden zur Wissensgewinnung zuständig. Von einem pädagogischen Standpunkt aus gesehen „setzt Lernanalytik dort an, wo Lerndesign endet“ [Lockyer und Dawson 2011].

Das Ergebnis der Analysemethoden gibt Auskunft über die Effektivität und Ausrichtung des Lernentwurfs. [Lockyer und Dawson 2011]

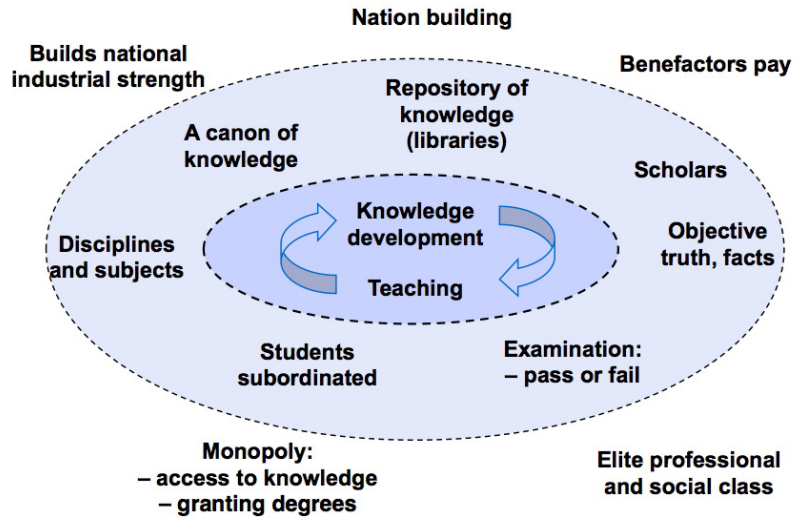
Abbildung 2.1 stellt ein Grundgerüst an Daten dar, welches deren Einsatzmöglichkeiten in einer Organisation zeigt. Die dreistufige Pyramide dient an der University of Phoenix als Ausgangsbasis dafür, wie dort Daten verwendet und analysiert werden. Die unterste Ebene umfasst Business Intelligence und Berichtswesen. Sie verwendet gesammelte Information zur Lösung einfacher Problemstellungen. Sie ist zudem für die Berichterstattung an etliche Geschäftsbereiche zuständig. Darauf aufgebaut ist die prädiktive Analytik, deren Methoden komplexere Aufgaben lösen und die anhand der Daten Lern- und Geschäftsvorhersagen trifft. Die oberste Stufe der Pyramide ist datengetriebenes Lernen. Erst hier wird Information – neben den Anwendungen der beiden anderen Ebenen – auf das Lernen selbst gebündelt und dahingehend eingesetzt. Oberstes Ziel ist die Hilfestellung für Studenten, deren Lernziele zu erreichen. [Sharkey 2011]



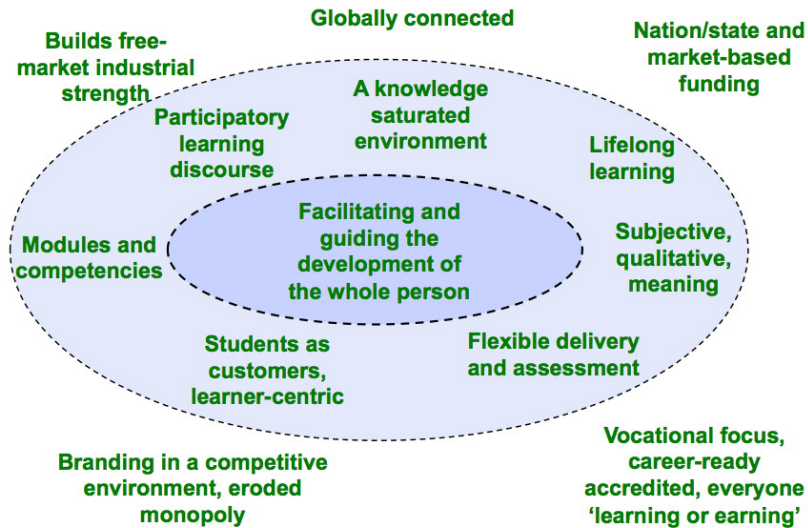
Abbildung 2.1: Pyramide zur Verwendung von Daten an der University of Phoenix [Sharkey 2011].

Neben dem technologischen Fortschritt und einem erkennbaren Wandel in der Gesellschaft, wo Wissenserwerb im Zuge des letzten Jahrzehnts eine bedeutende Rolle eingenommen hat, hat in Anbetracht der sich parallel dazu verändernden Art des Wissenserwerbs – des Lernens – eine Neugestaltung des Bildungssystems eingesetzt. Gemäß [Shum und Ferguson 2011] setzt sich diese Neustrukturierung der Bildungseinrichtungen (Universitäten, Schulen, et cetera) in den nächsten Jahren soweit fort, wie in Abbildung 2.2(b) zu sehen. Die Grafik zeigt am Beispiel Universität, welchen Herausforderungen Bildungseinrichtungen schon jetzt und in naher Zukunft gegenüberstehen werden, im Vergleich zu deren starrer Struktur vor dem sogenannten „Intensiv-Lernzeitalter“ [Miller und Bentley 2002], siehe dazu Abbildung 2.2(a). Im Mittelpunkt steht jetzt nicht mehr die Konzentration auf den reinen Wissenszuwachs, sondern vielmehr die Fokussierung auf die individuelle Ausbildung jedes Einzelnen. Globales Denken, die Erkenntnis der Notwendigkeit von lebenslangem Lernen und eine ständige Bereitstellung von Wissen haben die starren und lokalen Denkweisen, als auch objektive, unflexible Ansichten und das Lernen streng nach Vorgaben abgelöst. Es ist ersichtlich, dass die pädagogischen Anpassungen tief in die zugrundeliegende unzeitgemäße Bildungsstruktur eingreifen und sowohl Lerner als auch Lehrkräfte miteinbeziehen.





(a) Eigenschaften einer Universität im Industriezeitalter.



(b) Zukunftsweisende Eigenschaften einer Universität im „Intensiv-Lernzeitalter“.

Abbildung 2.2: Eigenschaften und Anforderungen an eine Universität im Laufe der Zeit nach [Shum und Ferguson 2011].

## 2.1 Learning Management Systems (LMS)

Online-Lernen beziehungsweise Online-Unterricht nimmt heutzutage einen immer größeren Stellenwert in unserem Bildungssystem ein. Zur Optimierung dieser Form des Lernens sowohl für Studierende als auch für Lehrende existieren sogenannte Individuelle Lernplattformen (ILP). Diese versuchen, sich den jeweiligen Eigenschaften der Anwenderin oder des Anwenders anzupassen. Eine große Rolle spielen dabei Einflüsse und Umgebung des Lerners sowie jene von Lehrenden, Fakultäten und Lernplänen. Ziel der Lernplattform ist neben der Zufriedenheit des Lerners die Verbesserung des individuellen Lernerfolgs. Untersuchungen, inwiefern der Einsatz von neuen Technologien in der Pädagogik Lernverbesserungen erzielt, werden in Methoden des Computer Supported Collaborative Learning (CSCL) zusammengefasst [Vatrapu et al. 2011]. Eine gewichtige Rolle für den Erfolg von Individuellen Lernplattformen stellt deren Kernkomponente dar, das Learning Management System (LMS). Es dient der eigentlichen Bereitstellung von Lerninhalten sowie der Interaktion zwischen Lernenden und Lehrenden. Weitere Aufgaben eines LMS sind Organisation und Verwaltung aller Lernaktivitäten sowie die Bereitstellung von Lernressourcen. Ein derartiges System sollte neben wissensfördernden Maßnahmen für Lerner nie die Lehrinhalte für Auszubildende außer Acht lassen, welche die Hauptbestandteile des Online-Lernens darstellen [Niemann et al. 2011]. Wie der bestmögliche Lernerfolg für den Lerner nun zustande kommt, hängt von den jeweiligen Lernmethoden und -modellen ab, mit welchen die genutzte Lernplattform arbeitet. Im Allgemeinen versucht ein derartiges LMS, den Lernfortschritt der Studenten zu überprüfen. Daraus schließt das System die weitere Vorgehensweise. Es kann die gesamte Aktivität des Lerners aufgezeichnet und bewertet werden. [Fancsali 2011]

Learning Management Systeme werden auf vielen Universitäten als Hauptantrieb für gemeinschaftliche Lernaktivitäten gesehen, besonders in Online-Netzwerken. Diese sind für pädagogische Zwecke von großem Nutzen. Demgegenüber genießt herkömmlicher traditioneller Unterricht noch immer den Vorteil, schneller und besser auf studentische Leistungen reagieren und in Folge Studenten passende Hilfestellungen anbieten zu können. Dieser Nachteil von Lernsystemen resultiert aus der noch zu ungenauen Interpretation all derjenigen Lernaktivitäten, welche online geleistet werden. Diesen wunden Punkt versucht die Soziale Netzwerkanalyse (SNA) durch Einsatz eines Frameworks auszumerzen (siehe Abschnitt 2.5). Beispiele für Learning-Management-Systeme sind Blackboard, Desire2Learn und Moodle. Letztgenanntes basiert auf Open Source. Die Entwicklung von LMS geschieht fast ausschließlich auf Benutzerseite, was zu Kompatibilitätsproblemen auf unterschiedlichen Systemen führen kann. Überhaupt gibt es noch keine Entwicklungsplattform auf Serverseite, welche dieses Problem eliminieren könnte. Ansätze zur Lösungsfindung findet man in zu definierenden Anforderungen zu Learning Tools Interoperability (LTI). LTI befindet sich noch im Entwicklungsstadium. [Bakharia und Dawson 2011]

Eine gewichtige Rolle wird der Personalisierung von Lernplattformen zugeschrieben. Dadurch können individuelle Lernstrategien angelegt und wichtige Lernunterlagen einfach zugänglich gemacht werden. Diese Entwicklung führt in weiterer Folge zu personalisierten Lernumgebungen, sogenannten Personal Learning Environments (PLE). [Fournier et al. 2011, Taraghi und Ebner 2012]

In rechnerübergreifender Lernanalytik werden Daten von Lernern automatisch gesammelt – mit dem Ziel, Lernfortschritte aufzuzeigen. Akademische Analytik verdeutlicht durch Kontrolle des Lerners und dahingehendes Eingreifen bei lernschwachen Teilnehmern zudem eine Steigerung der

persönlichen Lernbereitschaft und Leistung einerseits, sowie der eigenen Zufriedenheit andererseits. Feedback der Benutzer von Lernsystemen – sowohl der Studenten als auch der Lehrpersonen – ist ein wichtiger Faktor zur laufenden Verbesserung derselben. Das Projekt NEXT-TELL (Next Generation Education, Teaching and Learning for Life)<sup>1</sup> hat sich zur Aufgabe gemacht, Studenten und Lehrkräften durch neue und computergestützte Methoden Lernunterstützung zu liefern. Es wird von „neuen Anforderungen für Lehrkräfte im 21. Jahrhundert“ [Vatrapu et al. 2011] gesprochen. Im Mittelpunkt steht dabei die kontinuierliche Bereitstellung von erfassten Lerndaten, um personalisiertes und individuelles Lernen zu ermöglichen. [Vatrapu et al. 2011]

Der Versuch, visuelle Analytik mit Lernanalytik zu kombinieren, führt zum dreistufigen Modell der Lehranalytik (Teaching Analytics). Wie in Abbildung 2.3 zu sehen, besteht dieses Modell aus dem Zusammenspiel und gemeinsamen Wissenserwerb von Forschern (Design-Based Research Experts - DBRE), Lehrenden (Teaching Experts - TE) und Analytikern (Visual Analytics Experts - VAE). Durch den Einsatz von Techniken der visuellen Analytik werden Lernprozesse und Lernaktivitäten ausgewertet. Jeder Vertreter der drei genannten Stufen profitiert dabei von Rückmeldungen der zwei anderen, wodurch seine eigene Leistung positiv beeinflusst wird. Feedback von Lehrkräften führt zum Beispiel zu besserem Lehrverständnis der Forscher und neuer verbesserter visueller Technologien der Analytiker. All diese Rückmeldungen sollen schließlich zu einem für den jeweiligen Studenten passenden Feedback des Lehrenden zusammenlaufen. [Vatrapu et al. 2011]

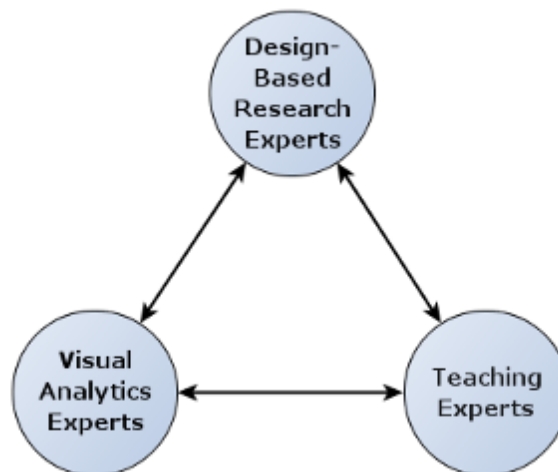


Abbildung 2.3: Dreistufiges Modell der Teaching Analytics nach [Vatrapu et al. 2011].

Das Zusammenspiel von Lernern, die Kommunikation untereinander, kennzeichnet einen wichtigen Faktor des Lernprozesses. Es wird versucht, die Interaktionen von Learning-Management-Systemen detailliert aufzuzeichnen. Die rasante Entwicklung des Webs macht das Erfassen dieser Informationen jedoch anhaltend schwieriger. Benutzern ist es in der Regel freigestellt, welche Kommunikationsmittel sie verwenden. Es muss nicht immer das interne Forum des Lernsystems sein, es gibt eine Vielzahl anderer Möglichkeiten. Der Nachteil davon liegt darin, dass das System für gewöhnlich nur die Aktivitäten innerhalb des Lernsystems abgleicht. Insofern können wichtige Informationen von Lernprozessen

<sup>1</sup> <http://www.next-tell.eu> [Zugriff am 21.08.2012]

(Interaktion, Lernressourcen außerhalb des LMS) verloren gehen. Einen Problemlösungsansatz stellt der Einsatz virtueller Maschinen dar. Besitzt eine virtuelle Maschine ein System zum Datenabgriff, so wird sichergestellt, dass beliebige Aktivitäten der Benutzer gespeichert bleiben. Die neu geschaffene virtuelle Umgebung kann innerhalb und außerhalb des Lernsystems auftretende Ereignisse erfassen. Bei Einsatz derartiger virtueller Maschinen muss auf Datenschutzbelange Rücksicht genommen werden. [Pardo und Kloos 2011]

### 2.1.1 Plattform für Lernanalytik

[Bader-Natal und Lotze 2011] beschreibt die Entwicklung einer Plattform für Lernanalytik in einem webbasierten Lernsystem namens Grockit<sup>2</sup>. Am Entwicklungsbeginn wurden fundamentale Fragen zu Aufgabenbereich und Zielsetzung der Applikation erörtert. Jede dieser Fragen stand letztendlich für ein eigenständiges Analysevorhaben und aus den Überlegungen konnte ein Grundgerüst gefunden werden. Es besteht aus einer automatisierten Datenerfassung und einer individuellen Datenauswahl. Letzteres soll für jede anfangs gefundene Frage die gewünschten Daten liefern. Dies geschieht meist mithilfe der Abfragesprache SQL (Structured Query Language). Ein wichtiger Pfeiler des Systems ist die Analyse der Daten, die sich der eigentlichen Beantwortung der Fragen kümmern kann. Die speziellen Daten dazu sind aus der vorher besprochenen Auswahlphase bereits vorhanden. Die Ergebnisse werden graphisch wiedergegeben und Analyse und Darstellung werden im Falle von Grockit mit R, einer Programmiersprache für Statistikprobleme, realisiert. Eine Verteilung der Analysresultate erfolgt systemintern an ausgewählte Anwender. Gibt es den Wunsch nach weiteren Untersuchungen, welche noch nicht umgesetzt wurden, so macht das Framework deren Durchführung problemlos möglich. Einzig die Datenerfassung und Analyseverfahren – das bedeutet, eine neue SQL- und R-Datei – müssen entworfen werden. Vergangene Analysen können für einen späteren Gebrauch gespeichert werden. Die Methode des Online Analytic Processing (OLAP) erlaubt der Analyseplattform beim Auftreten neuer und unterschiedlicher Daten dieselbe Fragestellung beizubehalten. Es bedarf keiner Änderung im Code, also der Analysedefinition, einzig die Daten dazu sind modifiziert. [Bader-Natal und Lotze 2011]

Die untersuchte Analyse-Anwendung für Grockit stellt Studenten und Dozenten laufend einen Bericht zu Arbeitsleistung und Lernfortschritt zur Verfügung. Für Studenten sind dadurch deren aktuelle Stärken und Schwächen auf einen Blick ersichtlich. Vortragende erhalten graphische Leistungsanalysen ihrer Studenten. In Abbildung 2.4 ist die Visualisierung eines derartigen Berichts für eine Studentin oder einen Studenten dargestellt. Die derzeitigen Lerndaten der Studentin beziehungsweise des Studenten werden angezeigt. Dazu gibt es die Möglichkeit, Einstellungen zu Schwierigkeit und Zeitbereich vorzunehmen. In diesem Zusammenhang, auch aufgrund der stetig steigenden Anzahl an Daten, darf nicht auf die Optimierung der Rechenleistung vergessen werden. [Bader-Natal und Lotze 2011]

### 2.1.2 Recommender Systems

Viele Learning-Management-Systeme greifen Daten zu den Lernaktivitäten ihrer Benutzer ab. Das führt zu einer ungeheuren Datenmenge, aus der oftmals versucht wird, geeignete Hilfsmittel zur Lernunterstützung für die Benutzer abzuleiten. Dies geschieht durch Ermittlung von Ähnlichkeiten zwischen Objekten, unter Anwendung passender Filtermethoden (zum Beispiel inhaltsbezogene Filte-

<sup>2</sup> <https://grockit.com/> [Zugriff am 21.08.2012]

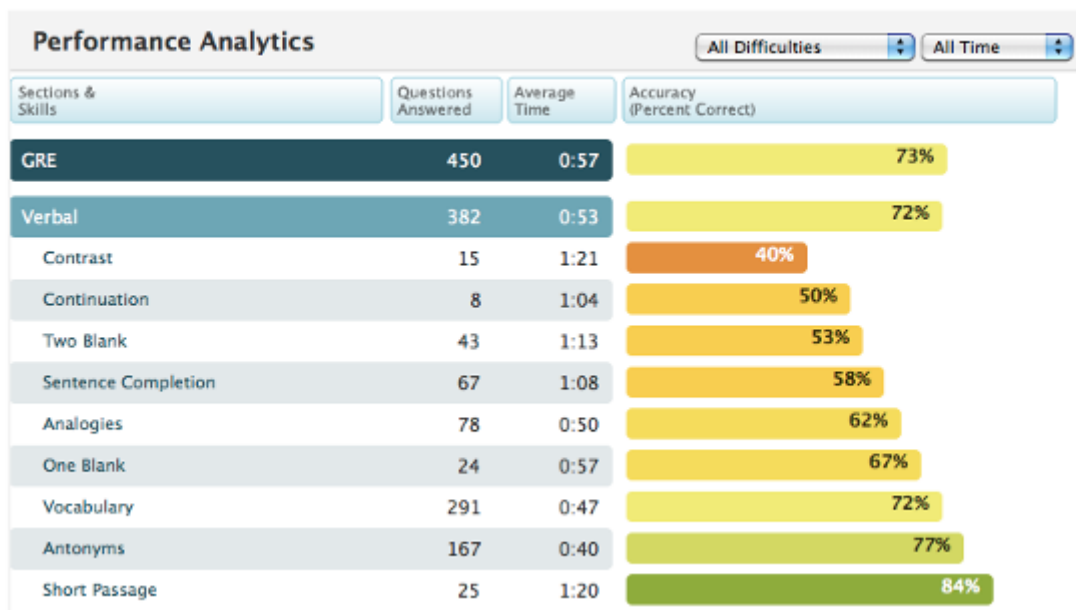


Abbildung 2.4: Performance-Report für eine Studentin beziehungsweise für einen Studenten gemäß [Bader-Natal und Lotze 2011].

nung, userbezogene Filterung). Ein Zusammenhang zwischen den resultierenden Empfehlungen und den aktuellen Lernaktivitäten des Users muss bestehen bleiben. [Niemann et al. 2011] untersucht in diesem Zusammenhang die Auffindung von semantischen Ähnlichkeiten von Lernressourcen durch Ermittlung der Kosinus-Ähnlichkeit zwischen den Dokumentvektoren der Ressourcen. Dokumentvektoren ergeben sich aus Metadaten wie Titel, Beschreibung, Speicherort und weiteren Eigenschaften von Lernquellen. Experimente bestätigen die These, dass diejenigen Ressourcen Ähnlichkeiten im Inhalt aufweisen, welche auch gleichartig von den Nutzern verwendet werden. Auch die Annahme, Lernquellen mit häufiger gemeinsamer Nutzung stünden in semantischer Beziehung zueinander, wird gestützt. Durch Untersuchungen am Einsatz von Lernressourcen können also Rückschlüsse auf Gemeinsamkeiten derselben getroffen werden. Neben Analysemethoden zur Bereitstellung von Informationen zum stetigen Lernfortschritt werden Untersuchungen zu früheren Lernprozessen immer wichtiger [Fournier et al. 2011]. Anhand dieser können wertvolle Empfehlungen zu Lerntechniken bereitgestellt werden [Niemann et al. 2011].

Empfehlungsdienste werden vielfach im Bereich E-Commerce eingesetzt. In lernbasierten Arbeitsgebieten – hier sei Technology Enhanced Learning (TEL) erwähnt – gewinnen derartige Systeme immer mehr an Bedeutung. Sie bieten Möglichkeiten zur Unterstützung für Lernende, aber auch für Lehrende. Zur Weiterentwicklung und Verbesserung von Recommender Systemen ist die Untersuchung von sogenannten TEL-Datensätzen maßgeblich. Diese Datensätze werden von unterschiedlichen Lernplattformen aufgezeichnet und bestehen neben der Identifikation der Anwender in der Regel aus User-Input beziehungsweise aus einfachem Feedback zu Objekten. Letzteres kann sich zusammensetzen aus Bewertungen und Nutzer-Kritik zu bestimmten Objekten, als auch aus verwendeten Suchbegriffen der Anwender. Auch implizites Feedback wie das Abgreifen der Lesedauer oder das

Speichern und Drucken eines Dokuments ist oft brauchbar. Der Auswahl an Daten sind hier keine Grenzen gesetzt, solange sie für den Empfehlungsdienst relevant sind, um daraus passende Lernvorschläge zu konstruieren. Für gewöhnlich werden frei verfügbare Datensätze von Lernumgebungen zur Evaluierung und Entwicklung von Empfehlungsdiensten und deren Algorithmen verwendet. Dieser Forschungsbereich hat in der jüngeren Vergangenheit einen Aufschwung erfahren, der bis in die Gegenwart reicht. Die relevanten Datensätze werden dazu öffentlich zur Verfügung gestellt und aktualisiert – auf der Website von TELEurope<sup>3</sup>. Der Datensatz der Forschungsplattform Mendeley ist einer davon. Er besteht aus einer Vielzahl von Informationen, mit denen Fachartikel und andere Dokumente als Lernquellen vorgeschlagen werden können. Beispiele für Systeme dieser Art, welche Dokumente empfehlen, sind InLinx oder Papyrus. Der Datensatz von Mendeley enthält Angaben zu allen verfügbaren Artikeln, deren Verwendung durch die Anwender sowie weitere Dateien zu User-Bewertungen und Kennzeichnung. Die Protokollierung geschieht seit dem Jahr 2009. Durch Einsatz dieses Datenbestandes können Anwenderinteressen und die Qualität einzelner Artikel untersucht und gezeigt werden. Ein brauchbares Beispiel zu gesammelter Lernleistung bietet der APOSDLE-DS Datensatz der Technischen Universität Graz<sup>4</sup>. Zahlreiche Lernaktivitäten von (jedoch nur sechs) Anwendern wurden protokolliert, darunter das einfache Betrachten von Lernressourcen oder die Kontaktaufnahme mit Personen. Für Architekturstudenten interessant ist der Datensatz MACE des MACE Projekts<sup>5</sup>. Er beinhaltet nützliche Log-Daten, von Suchbegriffen über Downloads bis hin zu Kommentaren und gesetzten Tags der Anwender. Deren Lernpräferenzen und Interessen können dadurch einfach ermittelt werden. Die angebotenen Lernmittel aus dem Bereich der Architektur verteilen sich auf ganz Europa. [Verbert et al. 2011]

Recommender Systeme arbeiten mit Algorithmen aus dem Gebiet der Datenanalyse. Wie vorher erwähnt, unterscheiden sich diese durch deren Filtermethoden. Die Methode des Filterns von gleich gesinnten Meinungen (Collaborative Filtering) ist die am besten entwickelte Technologie, die noch dazu am weitesten verbreitet ist. Inhaltsbezogene Filterung benutzt für die Berechnungen nur individuelle anwenderbezogene Informationen, während bei der hybriden Filterung mehrere Techniken gleichzeitig zur Auffindung von relevanten Empfehlungen im Einsatz sind. Die Methode des Kollaborativen Filterns versucht, aus speziellen Angaben beziehungsweise Bemerkungen von Anwendern zu einem Objekt – das können beispielsweise einfache Bewertungen sein – und anschließendem Vergleich mit den Daten anderer gleichgesinnter Benutzer, geeignete Empfehlungen zu berechnen. Diese Filterungstechnik kann wiederum in zwei Teilbereiche aufgeteilt werden, in Anwenderbasiertes Kollaboratives Filtern und in Objektbasiertes Kollaboratives Filtern. Welcher Algorithmus schlussendlich die geringste Fehlerstreuung ergibt, ist abhängig vom jeweiligen Datensatz. [Verbert et al. 2011]

### Anwenderbasierte Kollaborative Filtermethode

Algorithmen dieser Art versuchen, Ähnlichkeiten zwischen Anwendern zu finden – meist durch Untersuchung und Vergleich von deren Bewertungen an Objekten. Nach Auffindung von gleichartigen Benutzern können deren weitere Bewertungen als Annahme für Empfehlungen genutzt werden. In der Praxis werden dabei Benutzer (dargestellt als Vektor  $v$ ) anhand ihrer Bewertungsähnlichkeiten zum gerade aktiven User (dargestellt als Vektor  $u$ ) gewichtet. Die Ähnlichkeit zweier Benutzer kann durch

<sup>3</sup> <http://www.teleurope.eu/pg/pages/view/50630> [Zugriff am 21.08.2012]

<sup>4</sup> <http://www.aposdle.tugraz.at/> [Zugriff am 21.08.2012]

<sup>5</sup> <http://portal.mace-project.eu/> [Zugriff am 21.08.2012]

unterschiedliche Funktionen berechnet werden. Eine davon ist die Kosinus-Ähnlichkeit, welche sich aus nachfolgender Formel (2.1) ergibt.

$$w_{uv} = \frac{\sum_{i \in I_{uv}} r_{vi} r_{ui}}{\sqrt{\sum_{i \in I_{uv}} r_{vi}^2 \sum_{i \in I_{uv}} r_{ui}^2}} \quad (2.1)$$

In Formel (2.1) stellt  $I_{uv}$  den Objektsatz dar, welchen beide User bewertet haben,  $r_{ui}$  steht für die Bewertung des Objekts  $i$  durch Benutzer  $u$  und  $r_{vi}$  ist die Bewertung von Benutzer  $v$  für Objekt  $i$ .

Eine Gewichtung nach Ähnlichkeiten kann auch durch alternative Funktionen, wie die Pearson-Korrelation oder die Methode nach Tanimoto-Jaccard, ermittelt werden. Als Beispiel zeigen Experimente, dass bei kargen Datensätzen (z. B. MACE) die Anwendung von Tanimoto-Jaccard zu empfehlen ist. Nach der Berechnung derartiger Ähnlichkeitsgewichtungen ergibt sich der Algorithmus für die Erzeugung der voraussichtlichen Bewertung des aktuellen Users zu Formel (2.2). Dabei wird eine gewisse Anzahl an Benutzern mit größter Gewichtung als Experten  $\tau_u$  gekennzeichnet, die für Objekt  $i$  eine Bewertung abgegeben haben. Ein Testuser wird durch  $u$  definiert, ein Testobjekt durch  $i$ . [Verbert et al. 2011]

$$\hat{r}_{ui} = r_u + \frac{\sum_{v \in \tau_u} w_{uv} (r_{vi} - r_v)}{\sum_{v \in \tau_u} w_{uv}} \quad (2.2)$$

### Objektbasierte Kollaborative Filtermethode

Im Gegensatz zum vorangegangenen Unterkapitel 2.1.2 wird bei der Methode des objektbezogenen kollaborativen Filterns die Ähnlichkeit zwischen Objekten berechnet. Hauptsächlich geschieht dies durch Anlegen einer Matrix von Objekt-Paaren, welche die Zusammenhänge von Objekten darstellt. Gemeinsam mit den Anwenderdaten und Vergleichen mit deren Bewertungen zu ähnlichen Objekten werden Empfehlungen generiert. Die Berechnung der Ähnlichkeit zwischen zwei Objekten geschieht fast gleichartig wie jene zwischen zwei Benutzern. Auch hier können verschiedene Funktionen wie die Methode der Kosinus-Ähnlichkeit zur Berechnung herangezogen werden. Im Falle der Kosinus-Ähnlichkeit werden nun nicht mehr die Benutzer als Vektoren dargestellt, sondern die Objekte mit deren Bewertungen. Danach wird die daraus gewonnene Ähnlichkeit zwischen zwei Objekten dafür benutzt, die Vorhersage der Bewertung eines Objektes durch einen Benutzer zu generieren. Die Formel dazu berechnet die Vorhersage aus der gewichteten Summe aller Bewertungen des Benutzers  $u$  zu ähnlichen Objekten wie Objekt  $i$ , welche auch jeweils nach der Ähnlichkeit  $w_{ij}$  zwischen zwei Objekten  $i$  und  $j$  gewichtet werden. Sie ergibt sich zu Gleichung (2.3). [Verbert et al. 2011]

$$\hat{r}_{ui} = \frac{\sum_{\text{allsimilaritem}j} w_{ij} (r_{uj})}{\sum_{\text{allsimilaritem}j} w_{ij}} \quad (2.3)$$

## 2.2 Vorhersage- und Ursachenmodelle

Um Lernerfolge zu optimieren und in weiterer Folge innovative Systeme zu entwickeln, fokussiert [Fancsali 2011] auf der Bestimmung von Vorhersagen und Ursachen für Lernerfolge, positive wie negative. Dabei wird das Lernverhalten untersucht. Die Schwierigkeit liegt darin, bestimmte Merkmale zu filtern, welche für den Lernerfolg maßgeblich sind. Je früher diese erkannt werden, desto schneller kann in das Lernverhalten eingegriffen werden. Anders ausgedrückt, aufgrund des Auffindens derartiger Merkmale (= Ursachen) können Vorhersagen getroffen werden und anhand dieser Prognosen kann ein geeigneter Eingriff zur Lernverbesserung stattfinden.

Nur aufgrund von gesammelten Daten des Lernalgorithmus Rückschlüsse auf kausale Zusammenhänge zu schließen, ist schwierig. Eine Hilfestellung dazu liefern azyklisch gerichtete Graphen (DAGs - Directed Acyclic Graphs) – auch Bayes'sches Netz genannt. Sie stellen bestimmte hypothetische Variablen wahrscheinlichkeitsverteilt und deren kausale Beziehungen untereinander graphisch dar. Mithilfe von Bayes-Lernalgorithmen können weitere Analyseformen, wie zum Beispiel Pfadmodelle, entwickelt werden. Die Menge an abgegriffenen Daten eines Lernsystems ist sehr komplex und mehrdimensional. Enthaltene Variablen können einfach auszuwerten sein oder sich aus latenten Eigenschaften zusammensetzen. Andere wiederum existieren nur als Rohdaten und müssen einer Bearbeitung unterworfen werden, um Bedeutung zu erlangen. Durch dieses Neukonstruieren von Variablen wird die Dimensionalität der Daten herabgesetzt. [Fancsali 2011] beschreibt das Problem dahingehend, dass eine Menge von Variablen der Rohdaten  $\{R_1, \dots, R_n\}$  durch deterministische Funktionen  $\{f_1, \dots, f_k\}$  auf eine kleinere Variablenmenge  $\{C_1, \dots, C_k\}$  reduziert werden soll. Beispiele für konstruierte Variablen sind Prüfungsergebnisse oder Kursnoten, aus denen in weiterer Folge Vorhersagemodelle getroffen werden können. Diese neu erstellten Variablen sollen praktisch und aussagekräftig sein.

## 2.3 Diskursanalyse

[De Liddo et al. 2011] sieht die aktive Diskussion zwischen Studierenden als wichtigen Lernfaktor. Die Art des Diskurses lässt Rückschlüsse auf Interesse und individuelle Lerneinstellung zu. Analysen dieser pädagogischen Dialoge ergeben Hinweise und Gradmesser für Lernvorgänge. Maßgebend dafür sind die Art des Argumentierens, die sprachliche Ausdrucksweise, die Ideenfindung sowie die Erörterung von Unstimmigkeiten. Diskussion regt automatisch zum Mitdenken an, wodurch Lernen überhaupt erst möglich wird. Diskursanalyse bietet in diesem Sinne „ein besseres Verständnis des Lernprozesses“ [De Liddo et al. 2011]. Eine Bereicherung für die Analyse bieten Methoden der Argumentationstheorie.

Online-Lernsysteme stellen als Diskussionsmöglichkeiten Tools wie Wikis, Foren oder Blogs bereit. Das heißt, die sprachliche Komponente ist einer schriftlichen gewichen. Die Auswertung und Analyse der Dialoge muss sich dieser Änderung anpassen. Es werden in Textbeiträgen Schlüsselwörter und Lernzusammenhänge gefiltert und erörtert. Ein Analyse-Tool, welches sich diesen Aufgaben annimmt, ist beispielsweise die Webanwendung Cohere<sup>6</sup>. Angewendet auf zeitlich strukturierte Dialoge setzt Cohere alle getätigten Beiträge in semantische Verbindung zu einem Bezugsbeitrag. Dadurch ergibt sich ein klares Bild der Diskussion. Eine Zustimmung oder Ablehnung zu einem Beitrag ist

<sup>6</sup> <http://cohere.open.ac.uk/> [Zugriff am 21.08.2012]



sofort ersichtlich. Cohere ermöglicht zusätzlich die Anzeige der Dialogstruktur als semantisches Netzwerk. Neben dem Setzen von sinnvollen Zusammenhängen zwischen Postings markiert Cohere die einzelnen Beiträge mit Symbolen. Diese sollen die allgemeine Funktion des Beitrages widerspiegeln und auf den ersten Blick sichtbar machen. [De Liddo et al. 2011]

Lernanalysen haben unter anderem zum Ziel, den Lernfokus, die Interessen und die Lernbereitschaft von Studenten zu ermitteln. Dies geschieht durch Analysen aller Beiträge eines Benutzers. Besonders im Blickpunkt stehen dabei die Art der Beiträge – seine ‚Markierungen‘ – sowie deren Anzahl und Verschiedenheiten. Fragen, ob ein User selbst Ideen oder eigene Lösungsvorschläge zur Diskussion beiträgt, sich also Gedanken macht und zum Lernen beiträgt, oder ob man nur durch Fragestellungen auffällt, können somit beantwortet werden. Zusammenhänge zwischen Beiträgen eines Lerner in einer Unterhaltung und den Beiträgen anderer können grundsätzlich positiv (gleiche Meinungen), neutral (Beispielgebungen) oder negativ (andere Meinung) sein. Zusammengefasst in einer Tabelle können aus diesen sogenannten Verbindungstypen aussagekräftige Lerninterpretationen für Studierende abgeleitet werden. In Abbildung 2.5 ist eine Beispieldatenbank aller Verbindungsarten der Beiträge einer Studentin beziehungsweise eines Studenten in einer Konversation dargestellt. [De Liddo et al. 2011]

Link Types	
Name	Count
is an example of	42
builds on	17
is consistent with	17
responds to	11
uses/applies	10
challenges	10
evidence relating to	6
supports	6
is analogous to	6
illustrates	5
is inconsistent with	2
improves on	1
solves the problem	1
refutes	1
causes	1

**Legend:**

- Neutral link type
- Positive link type
- Negative link type

Abbildung 2.5: Mögliche Verbindungsarten in einer Konversation [De Liddo et al. 2011].

Durch die Bestimmung von Verbindungstypen ergibt sich zum Beispiel der Prozentsatz an positiven Beiträgen, also ein Hinweis auf positives Lernverhalten, zu Gleichung (2.4).

$$P[\%] = \frac{\sum_{g=1}^G N_g}{T} \times 100 \quad (2.4)$$

In Gleichung (2.4) stellt  $G$  die Anzahl der positiven (grünen) Verbindungsarten dar,  $N_g$  steht für die Häufigkeit der positiv verwendeten Links, und  $T$  kennzeichnet die Gesamtanzahl an Verbindungstypen. Weitere Untersuchungen von Link-Type-Tabellen (siehe Abbildung 2.5) beschäftigen sich mit der Ausdrucksweise in Dialogen. Unterschiedliche Rhetorik muss dabei nicht gleich unterschiedliche Meinung heißen. Die Schwierigkeit besteht darin, die Beiträge richtig – das heißt positiv, negativ oder neutral – zu deuten. Eine weitere Eigenschaft von Cohere im Bereich des ‚compared thinking‘ ist die Darstellung von Verbindungen zwischen gleichgesinnten Beiträgen. Deren Autoren werden ebenfalls angezeigt. Es wird dadurch möglich, auftretende Gruppen von Lernern und gleichdenkende Anwender zu identifizieren. Oftmals hilfreich für die Analyse ist das Visualisieren von Netzwerken semantischer Natur, was mithilfe von Anwendungen wie dem Analyseprogramm für soziale Netzwerke, UCINet<sup>7</sup>, oder NodeXL<sup>8</sup> möglich ist. [De Liddo et al. 2011]

Im Rahmen von Lernen ist gemäß [Ferguson und Shum 2011] derjenige Dialog zwischen Lernenden für Lernfortschritte maßgebend, welcher auf der Suche nach Aufschlüssen für Fragestellungen basiert. In Lernsystemen bilden Chat-Anwendungen den sprachlichen Dialog am besten ab. Die Autoren haben mit Hilfe des Web-Konferenz-Programmes Elluminate die Chatsessions aller Teilnehmer einer wissenschaftlichen Tagung mitprotokolliert. Die Analysen der Dialoge umfassten die Auffindung von bestimmten Schlüsselwörtern, welche für Erkundungsdialoge ausschlaggebend sind. Die Schlüsselwörter wurden im Vorhinein festgelegt. Diejenigen Beiträge mit untersuchungsrelevanten Daten wurden in eine Log-Datei gespeichert. Die gesammelten Informationen wurden danach ausgewertet. Verschiedenste Analysemethoden können dabei angewendet werden. Die Ermittlung der Durchschnittsmenge aller Erkundungsbeiträge pro Person oder der Mittelwert der geposteten Beiträge pro Minute sind nur zwei Beispiele für vielfältige Analysemöglichkeiten. Die Auswertung des Chats unter Berücksichtigung von Schlüsselwörtern erwies sich als genauer im Vergleich zu den einfachen generischen Analysen der Chat-Beiträge. Ein großer Vorteil für zukünftige Anwendungen wäre die Automatisierung dieser Analyseform. Die Diskursanalyse kann soweit gehen, dass die gespeicherten Schlüsselwörter für einen späteren Gebrauch gespeichert bleiben. Diese könnten in weiterer Folge von Systemen benutzt werden, welche Lerner zu diesen Inhalten führen, falls sie benötigt würden. Ein derartiger Ansatz führt zu Recommender-Systemen, welche in Abschnitt 2.1.2 beschrieben sind. [Ferguson und Shum 2011]

Untersuchungen zeigen, dass Lernmethoden und Eigenschaften von Lernern im Kontext verschiedener Kulturen variieren können. Man sollte daher beim Entwurf und Design von Lernsystemen sowie allgemein in der Lernanalytik immer darauf achten, dass es nicht einen einzig vorgegebenen Weg gibt, der zum Lernerfolg führt. Vielmehr sollte beim Aneignen von Wissen auf interkulturelle Verschiedenheiten Rücksicht genommen werden, um dem Auftreten von Schwierigkeiten im Lernprozess – ausgelöst durch ebensolche kulturellen Unterschiede – nicht von vornherein einen Spielraum zu geben. [Vatrapu 2011]

## 2.4 Verhaltensbeobachtung

Das Ziel einer jeden Lernumgebung sollte eine stetige Weiterentwicklung sein. Das ist nur möglich, wenn es laufend Rückmeldungen von Anwendern und Vortragenden zu Lernprozessen und Lernres-

<sup>7</sup> <https://sites.google.com/site/ucinetsoftware/home/> [Zugriff am 21.08.2012]

<sup>8</sup> <http://nodexl.codeplex.com/> [Zugriff am 21.08.2012]

sources gibt. Das Benutzerverhalten auf der Lernplattform spielt dabei eine wesentliche Rolle. Es enthält wichtige und vor allem eine Vielzahl an Daten über Interessen, Lernfortschritte und Leistung im Allgemeinen. Durch dieses Wissen kann zu gegebenen Zeitpunkten positiv in den Lernprozess eingegriffen werden. Dem Entwurf von Methoden zur Lernverbesserung sind hier keine Grenzen gesetzt. Oft genügt es, die Lernplattform den Interessen der Anwender entsprechend individuell anzupassen. Relevant für die Entwicklung von Lernsystemen ist darüber hinaus die Anwendung der *Formativen Evaluation* von Kursen. Sie trägt zu einer „Verbesserung der Effektivität und Effizienz von Lernressourcen“ [Graf et al. 2011] bei. Darüberhinaus soll diese Auswertungsmethode zur Förderung von Motivation und Lernengagement beitragen. [Richards und DeVries 2011] empfiehlt, dass *Formative Evaluation*, wenn möglich, vor Beginn oder während eines Kurses stattfindet. So kann auf positive sowie negative Ergebnisse von Lehrmethoden zum schnellst möglichen Zeitpunkt reagiert werden. Genauere Erkenntnisse werden erzielt, wenn nicht ein Kurs im Ganzen, sondern eigens für jede ausgeführte Lernaktivität eine *Formative Evaluierung* stattfindet. Es empfiehlt sich, die Ergebnisse der Analysen allen Benutzern zugänglich zu machen. Verbesserungen und Optimierungen als Resultat der Evaluierung sollten sofort durchgeführt werden. Weiters sollte versucht werden, Feedback über Lernaktivitäten in die jeweiligen Kurse beziehungsweise Abläufe einzubinden, da dadurch eine höhere Rückmeldungsquote garantiert wird. [Graf et al. 2011, Richards und DeVries 2011]

Im Zuge des Projekts „Moodle Analytics“ der Athabasca University wurde eine Web-Anwendung zum besseren Verständnis der Arbeit von Lernumgebungen entwickelt. Sie trägt den Namen Academic Analytics Tool (AAT), wird in Lernsystemen eingesetzt und ermöglicht den Abgriff und die Analyse von Verhaltensdaten seiner Benutzer. Das leistungsfähige Tool untersucht die Arbeit und Mitwirkung von Studenten in Online-Kursen, inwieweit Lernfortschritte erzielt werden können, und speichert detaillierte Daten dieser Beobachtungen ab. In weiterer Folge steht es Benutzern offen, aus der Reichhaltigkeit der Daten gewünschte Informationen zu filtern und diese einer individuellen Analyseform zu unterwerfen. Die Applikation ermöglicht ihren Anwendern, alle angebotenen Kurse in die Analysen einzubeziehen. Diese Besonderheit ist ein großer Vorteil im Gegensatz zu anderen verfügbaren Anwendungen, wo Kurse nur einzeln untersucht werden können. Das Academic Analytics Tool ist einfach erweiterbar und kann durch seine generische Programmierung in beliebigen Lernumgebungen eingesetzt werden. Die Anwendung greift zuerst auf die Datenbanken der Lernsysteme zu, um die gewünschten Daten abzugreifen. Danach finden individuelle Analysen darauf statt. Deren Ergebnisse werden in eigenen Datenbanken oder Dateien innerhalb des Tools abgespeichert. Zum Abruf der Informationen aus den Datenbanken der Lernsysteme sowie zur Auswahl der Analysen und weiteren Untersuchungen bietet die Applikation ein graphisches Interface an, welches die Wahl zwischen gewünschten Daten oder Methoden erleichtert. Ein mächtiges Analysewerkzeug sind vordefinierte und anwenderspezifische Muster an Daten, die gewünschte Informationen aus Datensätzen holen und verarbeiten. Auch die Verkettung von Mustern ist möglich. Dadurch können beliebige Analysen vollzogen werden, welche Rückschlüsse auf Lerntätigkeiten und Lernerfolg geben – zum Beispiel die Ermittlung von Prüfungsfragen, welche von Studenten am schwierigsten zu beantworten waren, oder die Untersuchung von wenig verwendeten Lernmaterialien, et cetera. Generell sollen mit Hilfe des Tools für Lernerfolg bewährte Lernobjekte identifiziert werden. Deren Einsatz kann in Folge ausgeweitet werden, währenddessen von wenig Erfolg gekrönte Lernobjekte Änderungen unterworfen werden müssen. Das alles führt zu einer laufenden Verbesserung von Online-Kursen. [Graf et al. 2011]

[Grau-Valldosera und Minguillón 2011] untersucht das Phänomen der Studienabbrecher an der Fernuniversität Oberta de Catalunya in Barcelona, Spanien. Hat ein Abbrecher vielfältige Gründe

für eine eigenständige Beendigung seiner Ausbildung (Familie, Arbeit, et cetera), sehen dies die Universitäten selbst als Fehler in deren Ausbildungsmethode. Ein Studienabbruch ergibt sich meist aus komplexen Gründen, die miteinander verflochten sind und eine Fortsetzung der Ausbildung als nicht sinnvoll erscheinen lassen. Nachforschungen zeigen, dass die Zahl der Abbrecher in Online-Lernumgebungen höher ist als jene in Universitäten ohne virtuellen Zugang. Man darf nicht außer Acht lassen, dass auch jene Studenten als Ausbildungsabbrecher behandelt werden, welche nur den schnellen Lernzuwachs ohne akademisches Abschlussziel suchen. Ein Grund für Studienabbrecher wird in der Schwierigkeit von Studienprogrammen gesehen. So gibt es beispielsweise in Studien der Ingenieurwissenschaften mehr Abbrecher als in anderen Studien. Angemerkt sei, dass dieses Faktum nicht auf Fernuniversitäten beschränkt ist. Die Untersuchung kommt zum Schluss, dass es keine eindeutige Definition von Studienabbruch gibt. Der Ansatz von [Grau-Valldosera und Minguillón 2011] versucht, für jedes Studienprogramm individuell die Anzahl an fortlaufenden Semestern zu ermitteln, in welchen Studenten nicht eingeschrieben sind, sie sozusagen eine Studienauserzeit nehmen. An der Fernuniversität in Barcelona wurden mit über zehnjähriger Laufzeit studienrelevante Daten gesammelt. Anhand deren Analyse konnte diejenige Anzahl an nicht registrierten aufeinanderfolgenden Semestern für einen Ausbildungsabbruch herangezogen werden, welche eine Wahrscheinlichkeit kleiner als fünf Prozent aufwies, dass das Studienprogramm doch noch fortgesetzt werden würde. So konnten unterschiedliche Definitionen für Studienabbruch (zum Beispiel ‚nach drei Semestern Studienpause‘) für spezifische Programme ermittelt werden. [Grau-Valldosera und Minguillón 2011]

## 2.5 Social Network Analysis (SNA)

Soziale Vernetzung hat sich im Laufe der letzten Jahre zu einem vertrauten Begriff entwickelt. Auch der Bereich des Lernens kann durch Einsatz von sozialen Netzwerken profitieren. In diesem Aspekt wird das Online-Netzwerk als ein ungezwungener Lernbereich angesehen, der aus einer Gruppe von Leuten besteht. Soziales Lernen ist seit jeher ein wichtiger Bestandteil der Pädagogik. Durch die totale Vernetzung kann nun Information auf schnellste Weise verbreitet werden. Eine Website, welche oben genannte Kriterien erfüllt, ist iSpot<sup>9</sup> und wird in [Clow und Makriyannis 2011] erwähnt. iSpot ist ein Beispiel für Lernen durch Mitwirken. Eine Verbindung zwischen formellem und informellem Lernen ist somit hergestellt. Ferner gibt es keine Vorteile oder Beschränkungen für bestimmte User und folglich werden beste Lernmöglichkeiten für alle garantiert. Die Anwendung überlässt es einzig und allein ihren Benutzern, Beobachtungen von Tier- und Pflanzenarten in der Natur auf der Website von iSpot zu dokumentieren. In weiterer Folge können andere Anwender diese Angaben auf ihren Wahrheitsgehalt hin verifizieren oder ausbessern. Benutzer erhalten dadurch Rückmeldungen zu ihren Beiträgen, welche honoriert werden können und gespeichert bleiben. Durch diese soziale Vernetzung entsteht ein durchdachtes Reputationssystem, welches sich als mächtiges Werkzeug für Nutzen und Motivation herausstellt. Es stellt eine Vertrauensbasis zwischen den Anwendern untereinander her. Lernen erfährt dadurch Unterstützung. Benutzer erhalten für Beiträge und Zustimmungen von anderen Reputationspunkte gutgeschrieben. Für den Lernprozess wesentlich ist die aktive Teilnahme im sozialen Netzwerk. Wichtig sind hier das Durchforsten und Lesen von Beiträgen, das eigenständige Kundtun von Rückmeldungen und Wissen, die Zusammenarbeit mit anderen, sowie die Kontrolle über den Inhalt. Der gegenseitige Informationsaustausch mit Lernern und Fachleuten ist für den

<sup>9</sup> <http://www.ispot.org.uk/> [Zugriff am 21.08.2012]

Lernfortschritt von wesentlicher Bedeutung. [Clow und Makriyannis 2011]

Durch Soziale Netzwerkanalyse (SNA) in Lernsystemen wird versucht, alle Aktivitäten eines Lernalters in seiner Lernumgebung zu erfassen und auszuwerten. Lehrenden soll dadurch unter anderem eine schnelle und angemessene Reaktion auf Lernprobleme seiner Studenten möglich gemacht werden. Die Daten des Lernsystems sollen im günstigsten Fall in Echtzeit bereitgestellt werden. Soziale Netzwerkanalyse versucht mit Hilfe eines Frameworks, die gesammelten komplexen Datensätze in Visualisierungen umzuwandeln. [Bakharia und Dawson 2011]

SNA dient zum besseren Verständnis von Lernerfolgen. Für Lehrende gibt die Analyseform seriöses Feedback, ob entworfene Lernmethoden von Erfolg gekrönt waren [Lockyer und Dawson 2011]. Ein Netzwerk im Bezug auf Lernen ist ein „offener Online-Raum“ [Fournier et al. 2011], in dem Benutzer kommunizieren, untereinander Ideen austauschen und verschiedenste Lernressourcen (Foren, Blogs, et cetera) benutzen. Durch Soziale Netzwerkanalyse kann dieses vernetzte Lernen analysiert werden. Zudem können alle relevanten Beteiligten im Lern-Netzwerk erfasst werden. [Fournier et al. 2011]

Soziale Netzwerkanalyse versucht, Interaktionsmuster und Zusammenhänge unter Lernern zu finden, um diese einer Untersuchung zu unterwerfen. [Teplovs et al. 2011] erwähnt die Implementierung eines Lerner Modells, welches sich durch Interaktionen mit anderen Lernern ergibt. Durch Soziale Netzwerkanalyse werden Gemeinsamkeiten von Anwendern gesucht und graphisch kenntlich gemacht („soziale Verbindungen“).

Ein brauchbares Tool, welches die Vorgaben für Soziale Netzwerkanalyse erfüllt, nennt sich Social Networks Adapting Pedagogical Practice, kurz SNAPP<sup>10</sup>. Es wurde 2008 entwickelt und ist als Browser-Erweiterung auf Benutzerebene verfügbar. Dessen Einsatzbereich sind Diskussionsforen, wo SNAPP in Echtzeit Soziale Netzwerkanalysen durchführt und diese sichtbar macht. In Lern-Management-Systemen gehören Diskussionsforen zu den am meisten benutzten Tools. Es werden interaktive Soziogramme präsentiert, welche von den Lernaktivitäten der Studierenden abgeleitet wurden. Soziogramme deshalb, da diese Art der Visualisierung eine einfache Interpretation der Daten für Lehrende zulässt. Beziehungen zwischen Nutzern ergeben sich aus deren Antwortbeiträgen im Diskussionsforum. In Abbildung 2.6 ist die Visualisierung eines Diskussionsforums durch das SNAPP-Tool zu sehen. Das Diagramm zeigt die Namen der Beitragsverfasser, die Verbindungen von Beiträgen zueinander und deren Richtung, sowie die Anzahl an gerichteten Postings. Rechts des Netzwerkdiagramms sind Elemente zur Steuerung und Filterung der Ausgabe angeordnet. Aus dem Diagramm ist ersichtlich, dass die Mehrheit der Knoten mit dem Knoten in der Mitte des Netzwerks verbunden sind und sich nach diesem richten. Dies stellt eine typische Schüler-Lehrer-Verbindung dar, bei welcher der mittlere Knoten die Lehrkraft charakterisiert. Wie zu sehen, hat die Lehrkraft die meisten Beiträge im Forum veröffentlicht und Antworten von mehreren Schülern darauf bekommen. Die Schüler untereinander haben kaum auf Beiträge anderer Schüler geantwortet. Diese Situation ist meist zu Beginn eines Kurses gegeben. [Bakharia und Dawson 2011, Merhbi 2011]

Neben Informationen zu Verfassern von Beiträgen und deren Antworten kann SNAPP den Inhalt von Diskussionen ermitteln [Fournier et al. 2011]. Ferner, ob deren Inhalt lernrelevant ist oder nicht. Ein großer Vorteil von SNAPP ist die zügige Bereitstellung von Informationen über die Aktivitäten und, in weiterer Folge, den generellen Lerneinsatz von Studenten. Dies ist ein wesentlicher Faktor

<sup>10</sup><http://research.uow.edu.au/learningnetworks/seeing/snapp/index.html> [Zugriff am 21.08.2012]



Lernen, im Gegensatz zum didaktischen Ansatz, der im klassischen formellen Lernen seine Ursache hat [Vatrapu 2011].

### 2.5.1 Das eTwinning-Netzwerk

Die rasante Ausbreitung des Web 2.0 und ihrer einhergehenden neuartigen Anwendungen wie beispielsweise Blogs oder Wikis eröffnet Schülern neue Lernerfahrungen und -möglichkeiten. Schwierigkeiten liegen darin, die Lernleistung des Einzelnen aus dessen Lernaktivitäten in diesen neuen sozialen Medien herauszufiltern. [Pham et al. 2012] erwähnen in diesem Zusammenhang das länderübergreifende Projekt eTwinning<sup>11</sup>, welches sich aus einem internationalen Zusammenschluss von 30 europäischen Bildungsministerien, dem European Schoolnet<sup>12</sup>, entwickelt hat. eTwinning ist ein europäisches Schulnetzwerk, welches Lehrpersonen neben einer länderübergreifenden Zusammenarbeit die Möglichkeit bietet, gemeinsame Projekte und Bildungsveranstaltungen auf internationaler Ebene durchzuführen. Der gegenseitige Austausch und die Zusammenarbeit sollen einzig und allein durch die Verwendung von Informations- und Kommunikationstechnologien über das eTwinning-Portal erfolgen. Die Plattform stellt dazu eine Reihe brauchbarer Dienste bereit. Schüler werden dadurch angeregt, sich in Projekten mit diesen modernen Technologien und Wissen auseinanderzusetzen. Das Erlernen dieser Fähigkeiten trägt zur Entwicklung der Medienkompetenz von Schülern bei und stellt eine gewichtige Hilfskomponente für das spätere Arbeitsleben dar. Ebenso werden Lehrkräfte auf dem eTwinning-Portal Lernern gleichgestellt, da sie ihre Kompetenzen durch aktive Mitarbeit an Projekten weiterentwickeln können und auch sollen. Derzeit – Stand Mai 2012 – gibt es 166.066 registrierte Benutzer, sogenannte eTwinners, und 5.353 aktive Projekte. [Pham et al. 2012, Education and Culture DG 2010]

Unter Zuhilfenahme von eigens entwickelten Analyseprogrammen können die gesammelten Daten des eTwinning-Netzwerks auf Besonderheiten und Zusammenhänge hin untersucht werden. Das Tool CAfe (Competence Analyst for eTwinning) sei hier als Beispiel genannt. Es analysiert Blogs und Projekte, um unter anderem die Lehr- und Kommunikationsfähigkeiten von Lehrenden zu beurteilen. [Pham et al. 2012] versuchen anhand von Methoden der Sozialen Netzwerkanalyse, spezifische Verhaltensmuster von Lehrkräften bei der aktiven Zusammenarbeit und Kommunikation im eTwinning-Netzwerk aufzuzeigen. Es zeigt sich, dass Lehrende Kontaktpflege und Projekte für wichtig halten und sich dementsprechend aktiv diesen Bereichen widmen, im Gegensatz zu Blognetzwerken, denen weniger Beachtung geschenkt wird. Die Untersuchung ergibt weiters, dass das Projekt Netzwerk eine solide Gemeinschaftsstruktur aufweist, die meiste Kommunikation also auf Basis von eTwinning-Projekten beruht.

## 2.6 Data-Mining

Die oft große Anzahl an Studenten verkompliziert Vortragenden das Erfassen aller studentischen Arbeitsvorgänge. Liegt ihr Hauptaugenmerk zudem nur auf den für die Notengebung relevanten ‚Endprodukten‘ wie Prüfungen oder Projektabschlüsse, und nicht auf den dazwischenliegenden Lernentwicklungen, so führt dies meist nicht zu den gewünschten Lernerfolgen. Der Einsatz von Techniken

<sup>11</sup> <http://www.etwinning.net/> [Zugriff am 21.08.2012]

<sup>12</sup> <http://www.europeanschoolnet.org/> [Zugriff am 21.08.2012]

des Data-Mining soll diesem Trend entgegenwirken und ist auf die Analyse von großen Datenmengen zugeschnitten. [Blikstein 2011] führt in diesem Zusammenhang die Begriffe Textanalyse und Emotionsermittlung ein. Durch Kenntnis der jeweiligen Stimmung von Studenten können detailliertere Rückschlüsse auf individuelle Lernfortschritte gezogen und außerdem feinfühligere Lernschritte gesetzt werden. Ein Beispiel dafür ist das Facial Action Coding System (FACS), welches den aktuellen Gemütszustand einer Person aufgrund seines Gesichtsausdrucks ermittelt. Zu einer Optimierung der Lernanalytik führt neben der Automatisierung der Analyse der (auch mehrdimensionalen) Daten die Erfassung aller Tätigkeiten, welche in Zusammenhang mit Lernen liegen. Dazu gehören neben anderen auch Gesten während des Lernens. Sie drücken die jeweilige Stimmung aus und liefern Rückschlüsse auf die Lernentwicklung. Das Fachgebiet Educational Data Mining (EDM) dient zum Messen des Lernfortschritts bei Studenten. Weitere Untersuchungen setzen im Bereich des Sprach- und Textmining an und führen bis hin zum Einsatz der Independent Component Analysis (ICA) Methode. Sie untersucht Unterhaltungen über Lerninhalte. Der Einsatz von Latent Semantic Analysis (LSA) gibt Rückschlüsse auf die Lernentwicklung und repräsentiert ein Modell für die Wissensgewinnung. LSA kann beispielsweise dann eingesetzt werden, wenn Ähnlichkeiten von Textbeiträgen gefunden werden sollen. [Blikstein 2011, Teplovs et al. 2011]

Für die Pädagogik besonders interessant erscheint die Fachrichtung des Educational Data Mining (EDM). Es befasst sich mit der Entwicklung von geeigneten Methoden der Datenauswertung zur Untersuchung und zum besseren Verständnis von Lernern und dessen Vorgangsweisen, um rückwirkend daraus lernsteigernde Verfahren zu erarbeiten. Eine entscheidende Frage ist dabei, wie Studenten ihren Lernvorgang bestreiten. Wesentlich erleichtert werden die Untersuchungen durch die Fülle an abgegriffenen Lerndaten aus Datenbanken und Lernsoftware, welche im Zeitalter von E-Learning in Genüge zur Verfügung stehen. Diese Daten unterscheiden sich je nach verwendetem Lernsystem. Learning Management Systeme liefern andere spezifische Lerndaten als Intelligent Tutoring Systeme (ITS) oder einfache Online-Prüfungen. Die anfangs unbearbeiteten Daten werden durch die Methoden des Educational Data-Mining in brauchbare Information umgesetzt, welche in weiterer Folge zur Verbesserung der Lernqualität und des Lernprozesses als Ganzes führt. Die gewonnene Information dient nicht nur Studenten und Lehrkräften, Rückschlüsse auf das Lernverhalten zu ziehen. Sie begünstigt auch andere Beteiligte in deren Wirken, wie etwa Universitäten bei der Kursauswahl oder Netzwerkadministratoren beim Einrichten von Lernsoftware, als Beispiele. [Romero und Ventura 2010]

Typische Data-Mining-Verfahren, welche bei Educational Data Mining häufige Anwendung finden, sind gemäß [Romero und Ventura 2010] die Regressionsanalyse, Clusteranalyse, Assoziationsanalyse und das Klassifikationsverfahren, während Entscheidungsbäume, Neuronale Netze und Bayes'sche Netze als klassische Methoden verwendet werden. Im Ausbildungsbereich besonders gefragt und gut untersucht sind Statistiken zu erbrachten Lernleistungen, Lernpräferenzen und gewöhnlichen Anwendungsdaten von Lernern. Die grafische Darstellung dieser Informationen ist hilfreich. Sie verstärkt die Aussagekraft wichtiger Daten, indem sie diese für die Anwender bildlich darstellt und dadurch auf einen Blick verständlich macht. Statistiken beruhen meist auf abgegriffenen Daten, welche in Log-Files oder Datenbanken gesammelt werden (siehe Abschnitt 2.7). Weitere maßgebende Anwendungen im Bereich des Educational Data Mining umfassen die Bereitstellung von Feedback für Lehrende sowie Systeme zur Empfehlung von geeigneten, meist persönlich angepassten Ressourcen oder Aktivitäten für Studenten. Die Vorhersage von studentischer Lernleistung stellt ebenfalls eine gewichtige Untersuchungsmethode dar, welche durch Dataming-Techniken erzielt wird. Da gegenwärtige Anwendungen für Lehrende großteils zu schwierig und vielschichtig zu verwenden sind,



sollte das Hauptaugenmerk bei zukünftigen Entwicklungen auf der Zunahme der Intuitivität und Automatisierung liegen. [Romero und Ventura 2010]

## 2.7 Logging

Vernetzte Lernumgebungen gehen immer mehr in offene Lern-Communities auf. Dort wird Teamwork großgeschrieben. Die Schwierigkeit beim Lernen besteht hier im „Zusammenspiel zwischen individueller und kollektiver Tätigkeit“ [Suthers und Rosen 2011]. Um verteilte Informationsbeschaffung überhaupt zu verstehen, sind etliche Untersuchungen zu Einzel- und Gruppenleistungen nötig. In vernetzten Lernumgebungen fokussiert die Wissensaneignung meist nicht auf einem bestimmten Medium. Vielmehr werden viele verschiedene Lernmittel zu Hilfe genommen, und seien es nur unterschiedliche Websites. Darüberhinaus existiert eine Menge neuer Medien wie Diskussionsboards, Chats oder Wikis, wo Information untereinander ausgetauscht und gelernt werden kann. Bestimmte Ereignisse oder Abläufe, welche sich in diesen Medien abspielen, können nun gefiltert und protokolliert werden. Es entstehen Log-Files, die abgespeichert werden, für Analysen aber meist noch ungeeignet sind, weshalb eine Umwandlung in andere Abstraktionsebenen von Vorteil ist. Eine Auswahl an geeigneten Abstraktionslayern zeigt Tabelle 2.1. [Suthers und Rosen 2011]

<b>Modell</b>	<b>Darstellungsform</b>
Process Trace	Log-Dateien, weitere Protokollierungsarten
Domain	Datensätze und deren Zusammenhänge
Event	Ereignissätze
Contextualized Action	Kontingenzgraphen
Interaction	Uptakegraphen
Mediation	Assoziogramme
Relationship	Untergraphen des Mediationsmodells
Tie	Soziogramme

Tabelle 2.1: Abstraktionsebenen analytischer Darstellungsformen nach [Suthers und Rosen 2011].

Den Anfang einer Interaktionsanalyse macht die Protokollierung aller Vorgänge, der Process Trace. Beispiele für diese oberste Ebene umfassen Log-Files, Audio- und Videoaufnahmen und Textprotokolle. Die Domain-Ebene bildet sich gemeinsam mit dem Event Modell aus den Log-Dateien. Sie beschreibt die involvierten Akteure in Verbindung mit den verfassten Events (zum Beispiel: Verfasser von Nachrichten, Nachrichten selbst, Anordnung nach Inhalt, et cetera). Events werden aus dem Process Trace durch bestimmte Analyseformen wie Exploratory Sequential Data Analysis (ESDA) gewonnen. Die zeitliche Zuordnung der Ereignisse führt zum Event Modell. Es beinhaltet die gesammte Erfassung der Events, deren Auslöser und Timestamp – wie im Log-File angegeben. Etliche Analysen können bereits auf diese beiden Ebenen angewendet werden. Dadurch sind fundamentale Informationen zu Teilnehmern und Inhalt jederzeit griffbereit. Setzt man die Events nun in Verbindung zueinander, ergibt sich die nächste Abstraktionsschicht mit Kontingenzgraphen. Diese stellen die Beziehung eines Events zu anderen Ereignissen dar. Dabei kann vom Autor selbst gewählt werden, welche Kontingenzarten (zum Beispiel Medienabhängigkeit) für die Analyse wichtig sind

und graphisch gezeigt werden. Das Zusammenspiel von Ereignissen führt zum Interaktionsmodell, das in Uptakegraphen dargestellt wird. Dabei wird sichtbar, wie Anwender von Ereignissen anderer – das können beispielsweise Beiträge sein – indirekt profitieren. Der Uptakegraph bildet sich vom Kontingenzgraphen heraus, wobei nur die für die Interaktion wichtigen Kontingenz-Informationen sichtbar bleiben. Dadurch ist eine exaktere Analyse möglich. Assoziogramme bilden die Relationen der Anwender untereinander ab, durch deren verwendeter Objekte. Die Graphen sind zweifach gerichtet (Aktoren - Objekte). Es entsteht das Mediationsmodell. Assoziogramme können aus Events tieferer Abstraktionsebenen (zum Beispiel aus dem Domain Modell) oder anderen Ereignissätzen gebildet werden. Sie dienen darüberhinaus als Hilfestellung, bestimmte Interessensmuster aus Kontingenz- oder Uptakegraphen zu filtern. Bildet man Untergraphen aller Pfade mit Länge zwei zweier Personen, das heißt es existieren direkte Verbindungen zwischen diesen Personen, so ergibt sich daraus das Relationship- oder Beziehungsmodell. Als letzte Abstraktionsstufe versucht das Tie Modell, Verbindungen zwischen Anwendern durch Soziogramme darzustellen. Verbindungen entstehen durch die vielschichtigen Interaktionen der Teilnehmer untereinander und können mit der Zeit komplexe Formen annehmen. Auf Soziogramme können Methoden der Social Network Analysis (SNA) angewendet werden, siehe Abschnitt 2.5. [Suthers und Rosen 2011]

Auf Programmiererebene sei die Umgebung NetLogo<sup>13</sup> erwähnt. Sie kann sämtlichen User-Input wie Mausklicks, Tastendrucke sowie Codeänderungen mitprotokollieren, indem die abgegriffenen Daten in eine XML Datei gespeichert werden. Mit Hilfe von Abfragesprachen wie XQuery werden die Informationen verarbeitet und analysiert. Ein sogenannter Code Navigator ermöglicht Forschern, die zeitliche Entwicklung von Programmen im Code vom Anfang bis zum Ende nachträglich zu begutachten. Diese beiden Tools lassen tiefgründige Codeanalysen zu und bringen individuelle Coding Strategien zum Vorschein. Abbildung 2.7 zeigt die Analyse eines Log-Files, welches während der Entwicklung eines kleinen Computerprogrammes angelegt wurde. Im Beispiel gut ersichtlich ist die zeitliche Variation der Zeichenmenge im Code. So gibt es im Anfangsbereich eine plötzliche Abnahme an Zeichen, was auf die Zuhilfenahme von Beispielmotiven hinweist, aus denen nicht gebrauchter Code sofort entfernt wurde. Nachträgliche Sprünge der Zeichenmenge lassen auf Wiederverwendung von eigenen Modulen im Code schließen. Gegen Ende der Entwicklung hat eine Formatierung des Codes stattgefunden. Mit dem Code Navigator wurden diese Schritte nachgeprüft und bestätigt. Es sei angemerkt, dass die Beispielabbildung nur einen Grundriss an Daten präsentiert, welche aufgezeichnet werden können. Derartige Untersuchungen zeigen, dass es grundsätzlich drei verschiedene Coding Profile gibt: Sogenannte ‚Copy and Pasters‘, Coder mit einer ‚Trial and Error‘ Strategie und eine Mischung davon. Das führt in Folge auch zu unterschiedlichen, für jedes Profil beziehungsweise für jede Coding Strategie eigens ausgelegte Anforderung für Lernunterstützung. Ein ‚Copy and Paster‘ braucht eine andere Hilfestellung als ein erfahrener Programmierer. [Blikstein 2011]

Die präzise Analyse von Chat-Protokollen stellt mitunter eine Schwierigkeit dar. Besonders in Umgebungen mit mehreren Benutzern (vergleiche Multi-User Umgebungen), wo aus Log-Dateien von Text-Chats Zusammenhänge aus einer Vielzahl von Beiträgen zu finden sind. Dabei stehen aufeinanderfolgende Beiträge eben nicht zwingend in Verbindung zueinander. Durch Einsatz eines individuellen Zeitfensters kann dieses Problem umgangen werden, wie [Rosen et al. 2011] erklärt. Dadurch werden nur diejenigen Beiträge untersucht, welche sich im aktuellen Zeitfenster befinden. Durch Analysemethoden können im Folgenden Ähnlichkeiten und Verbindungen zu anderen Beiträgen hergestellt werden.

<sup>13</sup><http://ccl.northwestern.edu/netlogo/> [Zugriff am 21.08.2012]

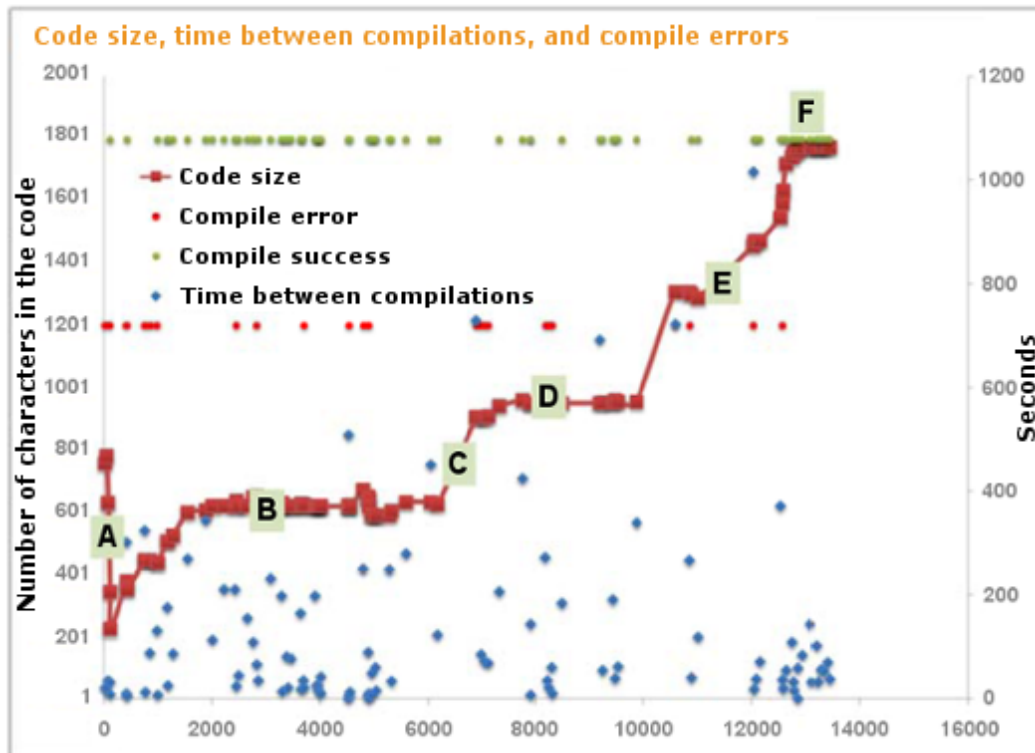


Abbildung 2.7: Beispiel zur Auswertung einer Log-Datei [Blikstein 2011].

### 2.7.1 Lecture Capture

Durch die fortschreitende Modernisierung rückt die Videoerfassung von Vorlesungen und deren nachfolgende Bereitstellung im Internet zunehmend in den Fokus von Lerninstitutionen. Unter einem Videoerfassungssystem im Lernbereich versteht [Brooks et al. 2011] die automatische Aufnahme, das Bearbeiten und das Zur-Verfügung-Stellen eines Beitrages für die Studenten. Ein Beispiel dafür ist das Recollect Lecture Capture System der Universität von Saskatchewan, Kanada. Es ermöglicht die Erfassung sowohl des Vortragenden selbst, als auch die Speicherung eines eigenen Audio- sowie Projektorsignals. Bestimmte Templates erlauben Vortragenden danach eine gewünschte individuelle Darstellung im Netz, das heißt sie können entscheiden, welches Signal wie in der Endfassung bei den Studenten am Bildschirm dargestellt wird.

Im Jahr 2010 wurde während eines vollen Semesters (15 Wochen) eine Studie in Auftrag gegeben, den Gebrauch des Recollect Systems zu dokumentieren. Ziel war es, neben der Protokollierung und Analyse von Lernaktivitäten, die Studenten anhand der erfassten Daten in Interessensgruppen zu unterteilen. Dafür protokollierte das System bei Verwendung sämtliche Aktivitäten des Users mit (zum Beispiel Klicks auf Play, Pause, Mausclicks, et cetera). Eine Clusteranalyse durch Einsatz des k-Means-Algorithmus und Weka Toolkits<sup>14</sup> hat ergeben, dass die Mehrheit der Studenten das

<sup>14</sup><http://www.cs.waikato.ac.nz/ml/weka/> [Zugriff am 21.08.2012]

Videoerfassungssystem wenig oder gar nicht benutzt hatten. Nur zeitlich kurz vor einer Prüfung in der Mitte des Semesters wurde das System außerordentlich oft genutzt. Gegen Ende der 15 Wochen, als die Endprüfung stattfand, gab es diesbezüglich jedoch keine Zunahme der Benutzerzahl, entgegen der Anfangsvermutung von [Brooks et al. 2011]. Zum Abschluss der Studie wurde eine Umfrage über den eigentlichen Nutzen des Systems durchgeführt. Hauptgrund für die Nichtbenutzung von Recollect war die grundlegende Einstellung der Studenten, das System biete keine zusätzliche Lernhilfe. Demgegenüber sahen die meisten der Studenten, welche das System verwendet haben, den Grund für die Zuhilfenahme von Recollect darin, dass dadurch keine zwingende Anwesenheit in der Vorlesung nötig war. Auch das nachträgliche Ergänzen von Mitschriften oder das Auffrischen des Verständnisses erschien den meisten Teilnehmern als wichtig. Als grundlegend negativ kann bei Benutzung des Systems das Hauptargument ihrer Nutzer betrachtet werden, nämlich die sinkende aktive Teilnehmerzahl in den Vorlesungen.

Die steigende Popularität der digitalen Aufzeichnung von Vorträgen hat Einrichtungen wie iTunes U<sup>15</sup> oder YouTube EDU<sup>16</sup> veranlasst, pädagogische Inhalte der Online-Welt zur Verfügung zu stellen. Einige Universitäten bieten in diesem Zusammenhang sogenannte OpenCourseWare an, das sind kostenlos ins Netz gestellte Kurse und Unterlagen dazu. [Brooks et al. 2011]

## 2.8 Kritische Stimmen zu Learning Analytics

Durch die steigende Popularität von Learning Analytics und dessen Untersuchungen werden auch vereinzelt kritische Stimmen laut. So können die pädagogischen Fähigkeiten von Lehrkräften nie mit künstlicher Intelligenz gleichgesetzt oder gar durch sie verdrängt werden. Vielmehr laufen Lehrende Gefahr, eine Abwertung ihrer Funktion und Vorbildwirkung zu erfahren. Lerner können ihrerseits fehlerhafte Systeme überlisten, indem gezielt nur diejenigen Kriterien eines Lernsystems erfüllt werden, welche für die finale Lernleistung – also die Benotung – maßgebend sind, während andere wichtige erzieherische Aspekte außen vor gelassen werden. Lerner werden beispielsweise durch das laufende Feedback des Systems in ihrer pädagogischen Entwicklung eingeschränkt und können ihr kognitives Wissen und andere Kompetenzen nicht ungehindert weiterentwickeln. Des Weiteren kann es vorkommen, bedeutungslose Leistungsbemessungen von Lernern zu erhalten. Für ein System ist es irrelevant, ob Schüler gezielt lernen oder nur vor sich hinspielen oder abgelenkt sind. Ebenso kritisch betrachtet wird die Verwendung, Speicherung und Auswertung der Lerndaten von Anwendern. Momentan sind der Nutzung sensibler oder privater Daten in diesem Bereich nur wenige Grenzen gesetzt. [Shum und Ferguson 2011]

<sup>15</sup> <http://www.apple.com/education/itunes-u/> [Zugriff am 21.08.2012]

<sup>16</sup> <http://www.youtube.com/education> [Zugriff am 21.08.2012]

### 3 Mathematische Lernsoftware

Es gibt eine Reihe von Software-Anwendungen, deren Zweck es ist, Information zu vermitteln und Lernen zu ermöglichen. In Schulen werden derartige Lern- beziehungsweise Übungsprogramme eingesetzt, um Schülern bestimmte Lerninhalte automatisiert näher zu bringen. Dieses Kapitel nimmt sich der Teildisziplin der Mathematik an und spezialisiert sich auf mathematische Lern- und Analyseprogramme. Bei der Verwendung von Lernsoftware kommt dem Erkennen und in weiterer Folge den Untersuchungen von Lernschwächen besondere Bedeutung zu. Auswertungen von Benutzereingaben und Analysen von Fehlern sollen helfen, Lernprobleme der Anwender aufzuzeigen und gleichzeitig auftretenden Lernschwächen entgegenzusteuern. Im Folgenden werden einige mathematische Lern- und Analyseprogramme beschrieben, welche individuelle Ansätze zur Fehlerdiagnose liefern. Bevor diese in den nächsten Kapiteln näher erläutert werden, sollen kurz und prägnant die Grundzüge und zwei Darstellungsmöglichkeiten einer Multiplikationsaufgabe gezeigt werden. Die nachstehenden Rechnungen (3.1) und (3.2) dienen als Exempel zur Erklärung aller beteiligten Faktoren einer Multiplikation.

$$241 * 32 = 7712 \tag{3.1}$$

$$\begin{array}{r} 241 * 32 \\ \hline 723 \\ 482 \\ \hline 7712 \end{array} \tag{3.2}$$

Die beiden Rechnungen in (3.1) und (3.2) stellen dieselbe Multiplikation dar, im dargestellten Beispiel  $241 * 32$ , gesprochen *241 mal 32*. Der äußerst linke Faktor 241 wird als Multiplikand der Multiplikation bezeichnet, welcher in diesem Fall aus drei Stellen besteht. Der zweistellige Faktor 32 der Rechenaufgabe stellt den sogenannten Multiplikator dar. Als Ergebnisprodukt beziehungsweise einfach als Produkt wird der Faktor 7712 bezeichnet. Das Produkt ergibt sich aus der rechnerischen Durchführung der Multiplikation von Multiplikand und Multiplikator. Die Darstellungsform der ersten Rechnung (3.1) stellt die Multiplikation und ihr Ergebnisprodukt in einfacher Weise dar, ohne die dafür benötigten Arbeitsschritte wiederzugeben. Diese sind in der zweiten abgebildeten Rechnung (3.2) zu sehen und stellen eine geläufige Übungsform für Schüler zum Erlernen der mehrstelligen Multiplikation dar. Sie wird als Schulmethode der Multiplikation bezeichnet [Eigenwillig und Mehlhorn 2008]. Die eigentliche Multiplikationsaufgabe ist dabei in der ersten Zeile abgebildet und durch einen waagrechteten Strich abgeschlossen. Die Zeilen darunter ergeben sich durch Multiplikation von jeweils einer Stelle des Multiplikators mit dem Multiplikanden. Eine der gängigsten Varianten ist es, mit der äußerst linken Stelle des Multiplikators zu beginnen, diese mit allen Stellen des Multiplikanden nach den

mathematischen Regeln zu multiplizieren, das Ergebnis anzuschreiben, und diese Abfolge bis zur äußersten rechten Stelle des Multiplikators zu wiederholen. Die Teilprodukte werden dabei stellenversetzt in Stufenform, wie in Darstellungsmöglichkeit (3.2) ersichtlich, angeschrieben. Sind alle Zwischenergebnisse angeschrieben, wird ein weiterer Strich darunter gesetzt. Als letzter Schritt werden die jeweiligen Teilprodukte spaltenweise, wie angegeben und unter Berücksichtigung vorkommender Überträge, addiert. Daraus ergibt sich der Faktor 7712 – das Endergebnis oder Ergebnisprodukt. Die Darstellung von Rechnung (3.2) kennzeichnet eine typische Übungsform von Lernprogrammen zur mehrstelligen Multiplikation, da alle wesentlichen Elemente zur Ergebnisfindung von den Schülern in ein derartiges vorgegebenes Formular eingetragen werden müssen und die Ermittlung von spezifischen Fehlern damit ermöglicht wird. Dieselbe Darstellungsform wurde auch bei der Entwicklung des rechnergestützten Multiplikations-Trainers angewendet, welcher in Kapitel 4 näher behandelt wird. Auch das dynamische Bewertungsprogramm DynaMath des nachfolgenden Kapitels 3.1 setzt auf diese Form der Zeilen- und Spaltendarstellung.

### 3.1 DynaMath

Bereits 1994 hat sich eine Gruppe von Professoren und Experten der Sonderpädagogik rund um Michael Gerber in [Gerber et al. 1994], welches die Grundlage dieses Kapitels darstellt, mit der Untersuchung von auftretenden Lernschwierigkeiten und einer rechnergestützten dynamischen Bewertung der mehrstelligen Multiplikation auseinandergesetzt. Zu jener Zeit haben Untersuchungen zur Sonderpädagogik den Fokus vermehrt auf die Gruppe der Lernschwachen gerichtet, ohne auf individuelle Bedürfnisse oder gar Gründe für auftretende Probleme und Schwächen einzugehen. Die Methode des dynamischen Bewertens von Lernern beziehungsweise von deren Lernleistung hat neue Möglichkeiten zur Problemlösung geliefert. Es werden dabei diejenigen Handlungen als wichtig angesehen und ermittelt, welche Einfluss auf den Lernprozess ausüben. Das sind beispielsweise simple Erklärungen oder Tipps von Lehrenden. Diese werden zur Leistungsfindung nun nicht ausgeklammert, sondern vielmehr und auch bewusst als wichtiger Teil dessen erkannt. Somit ermittelt die Methode der dynamischen Bewertung Abläufe, die selbst wesentlicher Bestandteil der zu bewertenden Leistung sind. Sie spezialisiert sich überdies auf eine individuelle Verbesserung der Lernleistung der Schüler.

In einer rund dreijährigen Entstehungs- und Testphase haben die Autoren mithilfe eines Programmierers ein rechnergestütztes Programm zur dynamischen Bewertung von mehrstelligen Multiplikationsaufgaben entwickelt. Es nennt sich DynaMath, später musste dieser Name aufgrund von Urheberrechtsbestimmungen auf DynoMath geändert werden. Im Folgenden wird das Programm mit der ursprünglichen Bezeichnung DynaMath angesprochen. Die Anwendung ist für Schüler mit Lernschwächen im Mittelschulbereich ausgelegt und untersucht die Leistung der Schüler während der Lösung von mehrstelligen Multiplikationen. Der Definitionsbereich für die Multiplikationsaufgaben ist dabei durch zwei- bis dreistellige Multiplizanden und ein- bis zweistellige Multiplikatoren eingegrenzt.

Wird das Programm gestartet, erfolgt für alle Schüler eine Vorabprüfung ihres mathematischen Wissens. Diese besteht aus allen 100 möglichen einstelligen Multiplikationen. Anhand der Ergebnisse des Vortests können Schüler individuell nach ihrem Können eingestuft werden. Die Autoren sprechen in diesem Zusammenhang von einem „Schwierigkeitsraum“ [Gerber et al. 1994], der sich für jeden einzelnen Schüler spezifisch ergibt und der durch die bewertete Eingangsüberprüfung aufgespannt

wird. Der Raum stellt die einstelligen Multiplikationen, geordnet nach bewerteter Schwierigkeit, dar. Dadurch ergibt sich pro Schülerin beziehungsweise pro Schüler eine individuelle Schwierigkeitsskala. In der informatischen Praxis wird dieser Schwierigkeitsraum durch ein Array repräsentiert, bestehend aus allen Multiplikationen des Eingangstests, welche nun nach deren erfasster Schwierigkeit angeordnet sind. Ein Algorithmus greift auf dieses Array zu und erzeugt in der Folge und anhand dessen interner Anordnung mehrstellige Multiplikationsaufgaben, welche im Laufe der Überprüfung von zunehmender Schwierigkeit sind. Die Schwierigkeit bezieht sich dabei einerseits auf die vorangegangene Einstufung der einfachen Multiplikationen der Vorabbewertung, hängt andererseits aber auch vom Komplexitätsgrad der Beispiele ab, der etwa durch Stellenzunahme einzelner Faktoren oder der Miteinbeziehung von Überträgen zunimmt. DynaMath untersucht jeweils zwei Aufgaben von Schülern. Auf Basis der eingegebenen Ergebnisdaten der Schüler berechnet der Algorithmus diejenigen beiden Multiplikationen als nachfolgende Aufgaben, welche die höchste Erfolgswahrscheinlichkeit aufweisen, um von den Benutzern gelöst zu werden. Eine nähere Erklärung zur Bildung der Aufgaben folgt in Unterkapitel 3.1.1.

DynaMath setzt sich aus einem Schüler-, Lehr- und Definitionsbereichsmodell zusammen. Ersteres dient zur Charakterisierung einer bestimmten Schülerin beziehungsweise eines Schülers mittels der individuell erbrachten Leistung. Diese wird durch ein individuelles Antwortprofil repräsentiert, welches regelmäßig aktualisiert wird. Die interne Darstellung erfolgt mithilfe einer Datenbank. Das Lehrmodell umfasst Hilfestellungen zu den Aufgaben, während sich das Definitionsbereichsmodell auf den Wissensbereich bezieht, der untersucht werden soll – in diesem Fall die Multiplikation. Der Definitionsbereich ergibt sich aus einer Vielzahl von Kombinationen von möglichen mehrstelligen Multiplikationen, die sich aus den nach Schwierigkeitsgrad geordneten einfachen Multiplikationen des Arrays der Eingangsbewertung ergeben. Hinzu kommt eine Unterteilung in vier festgelegte Kategorien von zunehmender Komplexität, welche in Tabelle 3.1 dargestellt sind. Die Variable  $n$  bezeichnet hier den Ziffernwert einer Zahl und kann einen beliebigen Wert von 0 bis 9 annehmen. Als Beispiel charakterisiert die Variablenfolge  $nn$  eine beliebige zweistellige Zahl. In Tabelle 3.1 nimmt die Komplexität der Multiplikationen mit steigender Kategorie zu, was gleichbedeutend mit einer Stellenzunahme des Multiplikanden und/oder des Multiplikators der Multiplikation ist. Aufgrund der individuellen Ergebnisse der Vorabüberprüfung ergibt sich so für jede Schülerin und jeden Schüler ein eigener Definitionsbereich.

Kategorie	Multiplikationsmuster
I	$nn * n$
II	$nnn * n$
III	$nn * nn$
IV	$nnn * nn$

Tabelle 3.1: Die vier Kategorien möglicher Multiplikationen mit steigendem Komplexitätsgrad [Gerber et al. 1994].

Die Autoren betonen, dass sich das Lehrmodell von DynaMath nicht mit den menschlichen Fähigkeiten einer Lehrperson deckt und auch nicht dahingehend verglichen werden soll, da ein Computerprogramm nicht in der Lage ist, die soziale Komponente von Menschen zu ersetzen. Noch weniger war das zu

Zeiten der Entwicklung des Programmes vor 20 Jahren möglich. Die Anwendung verfügt daher über ein eigenes Verfahren, um auf Fehler von Schülern bei der Ausführung oder auftretende Notwendigkeiten zur Hilfestellung zu reagieren. Man ließ dabei pädagogisches Wissen in das Lehrmodell einfließen, um eine möglichst realitätsgetreue Interaktion des Programmes mit den Schülern zu gewährleisten. Ein wesentlicher Teilbereich von DynaMath ist die Fähigkeit zur Fehlererkennung. Eine fehlerhafte Eingabe von Schülern wird erkannt, der Fehler selbst jedoch nicht als solcher kenntlich gemacht. Vielmehr wird das betroffene Eingabefeld unbefüllt angezeigt und es werden dazu individuell auf die Leistung der Schüler abgestimmte Hilfsmeldungen ausgegeben. Tritt ein Fehler erstmals auf, gibt das Programm neben einem akustischen Hinweis eine grafische Rückmeldung aus, welche auf die Stelle des Eingabefelders verweist. Wird trotz dieses Hinweises im nächsten Schritt erneut ein falsches Ergebnis von Anwendern produziert, werden das Problem und dessen zugehörige Komponenten analysiert und nachfolgend eine Abfolge von Hilfestellungen dazu ausgegeben, welche die Schülerin oder den Schüler zur Lösung des Fehlers führen sollen. Bleibt ein Fehler auch nach diesen beiden Hinweisstufen bestehen, werden allgemeine Arbeitsschritte und Techniken zur Lösung des betroffenen Problems in einer Animation vorgezeigt. Führt auch diese Maßnahme nicht zum Erfolg, werden die Lösungsschritte in einer ähnlichen Animation wie zuvor, nun aber explizit, demonstriert.

Das Programm wählt in dynamischer Weise diejenigen Benutzerhinweise zur Anzeige aus, welche für Schüler die höchste Erfolgswahrscheinlichkeit zur Problemlösung bieten und ordnet diese je nach Erfolgsausgang in einer Sequenz an. Während der laufenden Untersuchung richtet sich diese Anordnung daher immer wieder neu aus und verleiht DynaMath, hauptsächlich aufgrund dieser Eigenschaft, dessen dynamischen Charakter. Die Abfolge kann aus bis zu einer Menge von 19 verschiedenen Hilfestellungen bestehen und bildet gewissermaßen die psychologischen Eigenheiten von Schülern mit Lernschwierigkeiten ab.

### 3.1.1 Erzeugung der Aufgaben

Wie zuvor erwähnt, werden die Schüler anfangs einer Vorabüberprüfung über alle 100 einstelligen Multiplikationen unterzogen, deren Leistungsbewertung auf Genauigkeit und Geschwindigkeit erfolgt. Anhand der Resultate werden diese einfachen Multiplikationen pro Schülerin und Schüler nach steigender Schwierigkeit in einem Array sortiert. Es erfolgt eine Unterteilung der Anordnung in zehn Gruppen zu je zehn Multiplikationen. Beginnend mit der ersten und einfachsten Leistungsgruppe werden nun zufällige mehrstellige Multiplikationen aus dieser Gruppe gebildet. Anfangs Multiplikationen mit simplem Muster – vergleiche dazu Kategorie I aus vorangegangener Tabelle 3.1 – also Aufgaben bestehend aus einem nur zweistelligen Multiplikanden und einem einstelligen Multiplikator. Es werden alle Gruppen aufsteigend durchlaufen. Wurden Aufgaben aus allen zehn Leistungsgruppen erstellt, erhöht sich der Komplexitätsgrad der Beispiele, in diesem Fall auf Kategorie II (dreistellige Multiplikanden, einstellige Multiplikatoren). Die Generierung der Beispiele startet wieder aus der ersten Gruppe, ehe nachfolgend stufenweise die Schwierigkeit bis zur zehnten Gruppe erhöht wird. Dieser Prozess wiederholt sich, bis alle Komplexitätsgrade und Gruppen durchlaufen sind. DynaMath erzeugt jeweils zwei Problemstellungen pro Gruppe und Komplexitätsgrad. Nach Durchlauf aller zehn Gruppen ergeben sich also 20 Aufgaben, welche Schüler für eine bestimmte Multiplikationsstruktur zu lösen haben. Da der Definitionsbereich vier Kategorien von steigender Komplexität umfasst, erstellt das Programm alles in allem eine Menge von 80 Problemstellungen. Sie haben die Eigenschaft, dass jede dieser 80 Aufgaben, in aufsteigender Weise, eine individuelle und sukzessive Schwierigkeits-



zunahme für die Schülerin oder den Schüler darstellt. Es zeigt sich dabei, dass die Ergebnisse der Vortestung eine überaus gute Prognose für nachfolgende Leistungen an mehrstelligen Multiplikationen liefern. Um das Antwort- beziehungsweise Leistungsprofil der Schüler aktuell zu halten, werden zu jeder Beantwortung einer Aufgabe diverse Leistungsdaten wie Antwortzeit, Antwortgenauigkeit und Fehlereinstufung berechnet und gespeichert. Falls positive Antworten erst mithilfe von automatisierter Unterstützung zustande gekommen sind, erfolgt ebenso eine Speicherung von jenen Hilfestellungen, welche sich gerade als erfolgreiche Lernunterstützung für eine Schülerin oder einen Schüler bewiesen haben.

### 3.1.2 Bewertungsergebnisse

Die Autoren sehen die Ergebnisse des Bewertungsprozesses von DynaMath als eine Annäherung an die „Zone der nächsten Entwicklung (Zone of Proximal Development, ZPD)“ [Gerber et al. 1994] nach Vygotsky, welche die Abweichung von selbstständig erbrachter Leistung ohne Hilfe zu jener mit Hilfestellung darstellt. Dazu erstellt die Anwendung eine zweidimensionale Matrix bestehend aus 40 Elementen, in der jedes Element zwei Bewertungsaufgaben repräsentiert. Deren waagrechte Achse bildet die 10 Schwierigkeitsgruppen in ansteigender Form ab, während die vier Spalten der senkrechten Achse die Komplexitätszunahme der mehrstelligen Multiplikationen bezeichnen. Die Zone der nächsten Entwicklung umfasst nun genau jene Elemente der Matrix, deren Aufgaben einzig und allein durch Hilfestellungen des Programmes korrekt gelöst wurden. Diese Elemente sind in der Matrix farblich hervorgehoben und unterscheiden sich so von Elementen, deren Multiplikationen selbstständig gelöst wurden und jenen, deren Aufgaben trotz Hilfe falsch berechnet wurden. Je nach Ausrichtung und Eigenschaften der resultierenden Zone können Rückschlüsse und Interpretationen zur Leistung der Schüler und deren Rückwirkung auf erfolgreiche Unterstützung durch das Programm getroffen werden.

Zur weiteren Untersuchung der Bewertungsergebnisse der Schüler bietet DynaMath diverse Diagramme und Tabellen, welche aus den abgegriffenen Informationen in der Datenbank erstellt werden können. Zur Analyse können beliebige Größen herangezogen und dargestellt werden, wie beispielsweise die Anzahl an fehlerhaften Versuchen oder die für die Lösung eines Problems benötigte Zeit. Weiters wird die gesamte studentische Leistung aufgezeichnet, um sie danach in Echtzeit wiedergeben und analysieren zu können. Darüber hinaus generiert DynaMath anhand der gezeigten Leistung der Schüler einen schriftlichen Befund mit individuellen Lern- und Lehrempfehlungen. Pädagogen erhalten so eine Fülle an reichhaltigen Analysemöglichkeiten zur Leistungsuntersuchung.

## 3.2 TORUS

In diesem Kapitel werden, basierend auf [Woodward und Howard 1994], die rechnergestützte mathematische Anwendung TORUS und dessen Entwicklung erklärt. Zuvor werden innovative Ansätze der Lernforschung aus den 1990er Jahren, welche die Grundlage des Programmes bilden, aufgegriffen und behandelt. Gemäß Verhaltensforschern können beim Lernprozess auftretende Fehler mittels einer entsprechend gut gestalteten Unterrichtsform überwacht, beeinflusst und unterbunden werden. Beispielhaft für einen darauf angelegten und vorausschauenden Unterricht sind Lehrmethoden wie das unmittelbare Reagieren auf die Lernaktivitäten der Schüler in Form von Rückmeldungen, ein

durchdachtes Einbringen und Wiederholen von komplizierten Lernthemen sowie eine umsichtige Lehrplananalyse, neben anderen. Beruhend auf verhaltenstechnischen Untersuchungen in diesem Bereich führt ein derartiges Unterrichtskonzept für Schüler mit Lernschwierigkeiten zu einer Steigerung der Leistungsfähigkeit und Fehlerminimierung. Ein kognitiver Denkansatz verfolgt ähnliche Ideen zur Beseitigung von Lernfehlern und sieht gezieltes Üben und Nachprüfen als bedeutende Methoden dafür. Im Gegensatz zur Verhaltensforschung werden verfahrenstechnische systematische Fehler als schwer kontrollier- und korrigierbar gehalten und daher ein falsches Verständnis von Schülern als hartnäckig zu lösen angesehen. Erfahrungsgemäß versuchen Schüler, Fehler zu umgehen, indem bewusst andersartige Methoden oder Strategien zur Lösungsfindung gesucht beziehungsweise erdacht werden. Genau das ist jedoch Zeichen für ein vorliegendes Verständnisproblem. Missverständnisse dieser Art können oft über eine sehr lange Zeit bestehen bleiben. Dies erschwert eine nachhaltige Richtigstellung des Lösungsweges erheblich. Da eine genaue Analyse von Fehlern für Lehrkräfte zu zeitintensiv und zu komplex ist, wird dieser Vorgang auf rechnergestützte Diagnoseprogramme ausgelagert.

Oben erwähnte Überlegungen zur Leistungssteigerung durch Fehlerbeseitigung sowie das akademische Bedürfnis, Einblick in das mathematische Denken und Verständnis von Schülern zu erlangen, waren Antrieb für die Entwicklung eines zu jener Zeit innovativen rechnergestützten Bewertungsprogrammes namens TORUS. Ein Hauptaugenmerk bei dessen Entwicklung bildeten Aspekte der Sonderpädagogik, die gleichzeitig ein wichtiges Einsatzgebiet der Applikation sein sollte. Das Programm ist hauptsächlich für Schüler mit Lernschwächen beim Subtrahieren, mit Einschränkungen auch beim Addieren, ausgelegt. Die Aufmerksamkeit wurde auf die Subtraktion gelegt, da Untersuchungen gezeigt haben, dass diese Rechenart lernschwache Schüler vor deutlich mehr Probleme stellt als dies bei der Addition gegeben ist. TORUS beruft sich auf eine Anwendung mit dem Namen BUGGY und baut auf dessen wesentlicher Funktionalität auf. Bei BUGGY handelte es sich um ein Programm, welches seinen Ursprung in den späten 1970er Jahren fand und zur Auffindung von Fehlern bei Subtraktionsaufgaben gedient hat. Konkret hat sich die Anwendung auf die Ermittlung von fehlerhaften Arbeitsschritten beim Subtraktionsprozess konzentriert, um dadurch Fehlermuster ersichtlich zu machen. Es mussten dazu drei Fehler desselben Typs auftreten, um als Fehlermuster eingestuft zu werden. Spätere Erkenntnisse, wie etwa die Einsicht, dass Schüler gleiche Problemstellungen nicht unbedingt in gleicher Weise behandeln – dies macht die vorherige Ermittlung von Fehlermustern ungenau – führten zu Weiterentwicklungen des Programmes. Eine dieser überarbeiteten Versionen nennt sich IDEBUGGY und gilt als Vorläufer eines Intelligent Tutoring System (ITS). Die Verbesserung der Anwendung im Vergleich zu dessen Vorgängermodell bestand in der nunmehr dynamischen Erzeugung von individuell auf die Leistungskompetenz von Schülern ausgerichteten Subtraktionsaufgaben. Zu diesem Zweck wurden nach einer Reihe von sieben bis zehn Beispielen, welche von den Schülern abgearbeitet wurden, Fehlerhypothesen erstellt, anhand derer sich die nächste Problemstellung ergab. Je nach Antwort konnte so die aufgestellte Hypothese bestätigt oder widerlegt werden. Zunehmend hat sich gezeigt, dass nicht nur das Wissen über Lösungsverfahren von mathematischen Problemen von enormer Wichtigkeit ist, sondern auch das Verständnis der Ideen und Konzepte dahinter. Ein bedeutender Nachteil bei der Nutzung von BUGGY und dessen Nachfolgeprogrammen war zu jener Zeit die Eingrenzung der Fehleranalysen auf das Programm selbst. Weder im Umfeld des Lehrenden noch der Schüler fanden tiefgreifende Untersuchungen zu möglichen Fehlerquellen statt, den Gründen für das Auftreten von Irrtümern – etwa unklare Lehrpraktiken oder missverständliche Lernmaterialien – wurde schlicht nicht nachgegangen. Diese Einschränkungen zu beheben, wurde zu einer der Hauptaufgaben für die Entwickler des TORUS Systems.

Das Bewertungsverfahren des TORUS Programmes spaltet sich in zwei grundsätzliche Teile auf. Der erste Part umfasst eine schriftliche Überprüfung über 25 Subtraktionsbeispiele, die für Schüler der Mittelstufe (etwa zehn bis 13 Jahre alt) mit Lerneinschränkungen ausgelegt sind. Die Testung erfolgt schriftlich, da zu jener Zeit der Umgang mit Computern keinesfalls alltäglich oder vertraut war und die elektronische Eingabe genau aus diesem Grund zu ungewollten Fehlern führen konnte. Dies wird nach Abschluss der Überprüfung von der Lehrkraft selbst oder einer anderen Person erledigt. Danach untersucht TORUS die Antworten auf spezifische Fehler und Muster von Fehlern, indem alle Beispiele inklusive Lösungsversuchen gegenübergestellt und miteinander verglichen werden. Für den Fall, dass eine Schülerin beziehungsweise ein Schüler alle Aufgaben fehlerfrei gelöst hat oder das Programm alle vorkommenden Fehler zweifelsfrei erkannt hat, wird dieses Ergebnis der Lehrkraft zugänglich gemacht und die Untersuchung des Programmes gilt für die Schülerin oder den Schüler als beendet. Kann die Anwendung keine exakte Leistungs- beziehungsweise Fehlerdiagnose stellen, so tritt der zweite Teil des Bewertungssystems von TORUS in Kraft – Schüler werden einer weiteren Überprüfung über fünf Subtraktionsaufgaben unterworfen. Diese ergeben sich anhand von Analyseprozessen der ersten Phase und sollen dort gefundene Hypothesen über möglicherweise vorherrschende Verständnisprobleme bei den Schülern bestätigen oder widerlegen. Wie zuvor erhalten die Schüler die Beispiele in schriftlicher Form.

Analyseprogramme sind in der Lage, gute Prognosen für Leistungsdarbietungen von Schülern und das Auftreten von systematischen Fehlern zu treffen. Eine Schwierigkeit stellt das Erkennen und Bewerten von Schlampigkeitsfehlern dar, die in zufälliger Weise auftreten können und typischerweise durch Leichtsinn, Müdigkeit oder Ablenkung entstehen. Um derartige Fehler einfacher bestimmen zu können, verfügt TORUS über ein umfangreiches Diagnosesystem zur Ausweitung der Leistungsuntersuchung bei Schülern. Neben immer wieder auftretenden Subtraktionsfehlern kann das Programm systematische Fehler bezogen auf bestimmte Problemstellungen feststellen. Zusätzlich stellt das Tool diverse Auswertungsergebnisse in anschaulicher Form bereit. Die Algorithmen der Anwendung zur Fehlersuche lehnen sich an Literatur und Praxis an. Zum einen werden diejenigen Fehler berücksichtigt, welche anhand von Literaturrecherchen als die am meist verbreitetsten gelten. Zum anderen stützt sich TORUS auf typische Fehler, welche das Resultat einer mathematischen Untersuchung an mehr als 300 lernschwachen Schülern im Sonderschulbereich waren. Das Programm wurde auf mathematisches Fachwissen abgestimmt und nutzt ein sogenanntes Expertensystem zur Ermittlung und Anpassung seiner Diagnoseverfahren. Die Verwendung einer Datenbank gewährleistet die Sicherung aller relevanten Leistungsdaten.

### 3.3 TutorIT

Dieser Abschnitt behandelt eine zeitgemäße Lernanwendung, das universelle mathematische Unterrichtsprogramm TutorIT, erläutert dessen Grundlagen und beruht auf [Scandura 2011]. Scandura spricht darin die nicht aufzuhaltende Automatisierung des menschlichen Handelns an. Aufwendige Arbeitsprozesse werden dabei soweit automatisiert, dass deren rechnergestützte Abarbeitung schneller und fehlerfreier ist, als dies durch menschliches Tun möglich wäre. Es ist genau jener Ansatz der Automatisierung, der einen wesentlichen Bereich des TutorIT-Systems kennzeichnet und es so zu einem mit vielen Vorteilen bedachten Unterrichtsprogramm macht.

In diversen Lernprogrammen findet man unterschiedliche Ansätze zur Lösung von pädagogischen Problemstellungen. In dieser Hinsicht relevant ist einerseits das Bereitstellen von Hilfestellungen zur Problemlösung, sogenanntes *Scaffolding*, sowie andererseits der Zugriff auf elementare Grundlagen als Unterstützung zur Lösungsfindung. Rechnergestützten Unterricht dieser Art bezeichnet man als Computer Based Instruction (CBI). Dessen Entwicklung setzte bereits in den 1960er Jahren ein. Die generelle Ermittlung, inwieweit Lernsysteme von pädagogischem Nutzen sind, ist meist schwierig und teuer. Daher wird der Blick auf Entwurfsrichtlinien und Designmodelle gelenkt, welche als Grundlagen für einige gegenwärtige Lernprogramme dienen. Doch ist auch der Bereich der Computer Based Instruction mitunter nicht frei von Schwierigkeiten. Neben einer unbeweglichen Unterrichtsform stellt der Übergang von theoretischer Wissensdarstellung zu praktisch darauf aufbauenden Systemen eine weitere Hürde dar – passende Basismodelle zu finden, ist nicht immer einfach.

Weiterentwicklungen im Gebiet des computergestützten Unterrichts führten zur Entstehung von Intelligent Tutoring Systems (ITS). Derartige Systeme sind darauf ausgerichtet, während des Unterrichtsablaufes die Leistung von Schülern auf Defizite und Fehler hin zu untersuchen und infolgedessen Entscheidungen zum weiteren Geschehen zu treffen. Es ist dieser dynamische Eingriff in den Lernprozess, der Intelligent Tutoring Systems nahe an die Handlungsweise von Lehrpersonen in der Praxis heranführen lässt. Schwierigkeiten bei der Bestimmung von geeigneten Maßnahmen zur Verbesserung der Lehre in Lernsystemen liegen – neben dem Versuch, Annahmen über die gedanklichen Vorgänge von Lernern zu finden – hauptsächlich in der Frage, *was* gelehrt werden soll. Meist beruht diese Frage einzig und allein auf dem gerade behandelten Lehrstoff, da nur basierend auf dieser Information präzise Entscheidungen zu inhaltlich passenden Lehrmaßnahmen getroffen werden können. Im Gegensatz zu Intelligent Tutoring Systems stellt die Methode des Constraint Based Modeling (CBM) nicht kognitive Lehrpraktiken in den Mittelpunkt, das Hauptaugenmerk wird ausschließlich auf gewisse Bedingungen gelegt, die während des Unterrichts erfüllt werden müssen.

Erst die Entwicklung von Systemen wie AuthorIT und TutorIT sollte dem Autor zufolge die hoch gesteckten Ziele von ITS-Anwendungen erfüllen, nämlich mitunter besser und erfolgreicher zu lehren als dies durch Menschen möglich wäre. Beide erwähnten Systeme bauen auf Methoden der Strukturellen Lerntheorie (Structural Learning Theory - SLT) nach Scandura auf. Diese wurde ab 1970 erforscht und zielt darauf ab, menschliches Erkenntnisvermögen und Verhalten mit inhaltlichen Strukturen zu verbinden. Ein Hauptaugenmerk liegt dabei wieder auf der Frage, welche Eigenschaften und Inhalte vor allem in schwierigen pädagogischen Bereichen als unumgänglich erachtet werden, um mit Erfolg bedachtes Lernen zu gewährleisten. Die Art der Wissensdarstellung in der Strukturellen Lerntheorie wird als richtungweisend für die Entwicklung von leistungsfähigen Lernsystemen wie TutorIT angesehen. Eine bedeutende Rolle zur Abbildung von Wissen nimmt dabei die verfahrensorientierte Darstellung durch abstrakte Syntaxbäume (Abstract Syntax Trees - ASTs) ein, die gerichtete Graphen als frühere Repräsentationsform abgelöst haben. Abstrakte Syntaxbäume ermöglichen unter anderem eine „simultane Wissensdarstellung auf mehreren Abstraktionsebenen“ [Scandura 2011] sowie eine automatische Erzeugung von Testaufgaben und deren Lösungen. In dieser Hinsicht ebenso maßgeblich ist die Methode der Strukturanalyse (Structural Analysis - SA), die lernrelevante Handlungsweisen erkennt, diese in ihre Einzelteile auflöst und sie dann in eine hierarchische Form bringt, angeordnet nach Wissensvoraussetzungen, welche jeweils die Grundlagen zur Lösung von übergeordneten Problemen bilden. Anders ausgedrückt ermöglicht diese Methode eine Bestimmung davon, was genau Schüler lernen müssen und wie detailliert sie sich damit

auseinandersetzen müssen, um während des Lernvorgangs mit den angesprochenen Wissensgrundlagen in Verbindung zu kommen – anders ausgedrückt, um erfolgreiches Lernen zu gewährleisten. Durch die hierarchische Anordnung kann der aktuelle Wissensstand von Schülern jederzeit ermittelt und dadurch der weitere Unterricht gezielt beeinflusst werden. Kann eine durchgehende Untersuchung oder vollständige hierarchische Unterteilung nicht durchgeführt werden, so sei angemerkt, dass auch unvollendete Analysen erste wichtige Anhaltspunkte zur Wissensermittlung liefern können.

Scandura sieht einen entscheidenden Grund für den relativ geringen Bestand an adaptiven Tutoren-Systemen in der Schwierigkeit und Kostspieligkeit von deren Entwicklung. Probleme treten großteils bei der Ermittlung und Behandlung von pädagogischen Erkenntnissen und daraus abgeleiteten Eingriffen in den rechnergestützten Unterricht auf. Das angesprochene TutorIT-System tritt diesem Manko durch die Automatisierung des bedeutenden Verfahrens, welches die pädagogischen Befunde berechnet, entgegen. Diese Eigenschaft führte zu einer wesentlich vereinfachten und kostengünstigeren Entwicklung.

Bei TutorIT Math<sup>1</sup>, kurz TutorIT, handelt es sich um ein Tutoren- oder Unterrichtssystem, welches eindeutiges Wissen über eine Vielzahl von hauptsächlich mathematischen Problemstellungen – im Programm als Tutorials bezeichnet – zur Verfügung stellt. Eng damit verknüpft ist das Autorensystem AuthorIT, das die Entwicklungsgrundlage für angebotene Information in TutorIT ist. Es setzt sich aus drei Hauptkomponenten zusammen – dem AutoBuilder, Blackboard Editor und einem Auswahlwerkzeug. Der AutoBuilder ist für die Informationsdarstellung zuständig. Diese erfolgt verfahrensorientiert und wird intern durch abstrakte Syntaxbäume realisiert, welche im AutoBuilder als sogenannte Flexforms, einer XML-nahen Darstellungsform des Programms, wiedergegeben werden. Eine weitere wichtige Funktion dieser Komponente ist es, für die Anwender passende Aufgabenformulierungen, Hilfestellungen und positive sowie negative Rückmeldungen zu Problemlösungen bereitzustellen. Darüber hinaus kann die Genauigkeit der Informationsdarstellung definiert werden. Dies bildet gleichbedeutend den aktuellen Kenntnisstand von Schülern ab, der dadurch jederzeit bekannt ist.

Die Benutzerschnittstelle von TutorIT ist durch das Blackboard Interface implementiert. Im gleichnamigen Editor können dazu Vorlagen für Aufgabenstellungen auf individuelle Art und Weise angelegt und gestaltet werden. Das Unterprogramm besteht aus einem zentralen Hauptfenster, in dem Aufbau und Anordnung des Beispiels stattfinden. Außerdem können hier vorher genannte Elemente wie Hinweise oder Feedback wie gewünscht platziert werden. Zur Ausgestaltung der Darstellungselemente mit Animationen, akustischer Untermalung, individuellen Eingabefeldern und weiteren Merkmalen, steht ein Menü bereit.

Das Auswahlwerkzeug bietet die Möglichkeit, unterschiedliche Betriebsarten und Lernmodi einzustellen. Die Benutzer haben die Wahl zwischen den fünf Verfahren *Adaptive*, *Instruction*, *Diagnostic*, *Simulation* und *Practice*, die nach Auswahl jeweils weitere individuelle Konfigurationsmöglichkeiten bieten. Demnach können diverse Pädagogikansätze einander gegenübergestellt werden. Abbildung 3.1 zeigt das Auswahlinstrument des Systems im eingestellten *Adaptive* Modus mit dessen Vielzahl von Unteroptionen, die für dieses Verfahren bereit stehen. Beispielsweise stellt das Programm im gewählten Modus (sprich *Adaptive*) den jeweiligen Wissensstand des Lernalters abschnittsweise zur Verfügung.

AuthorIT arbeitet also im Hintergrund an der Erstellung und Behandlung der Problemstellungen, während das TutorIT-System dem Lerner die erzeugten Aufgaben weiterleitet, präsentiert und dessen

<sup>1</sup> <http://www.TutorITmath.com> [Zugriff am 21.08.2012]

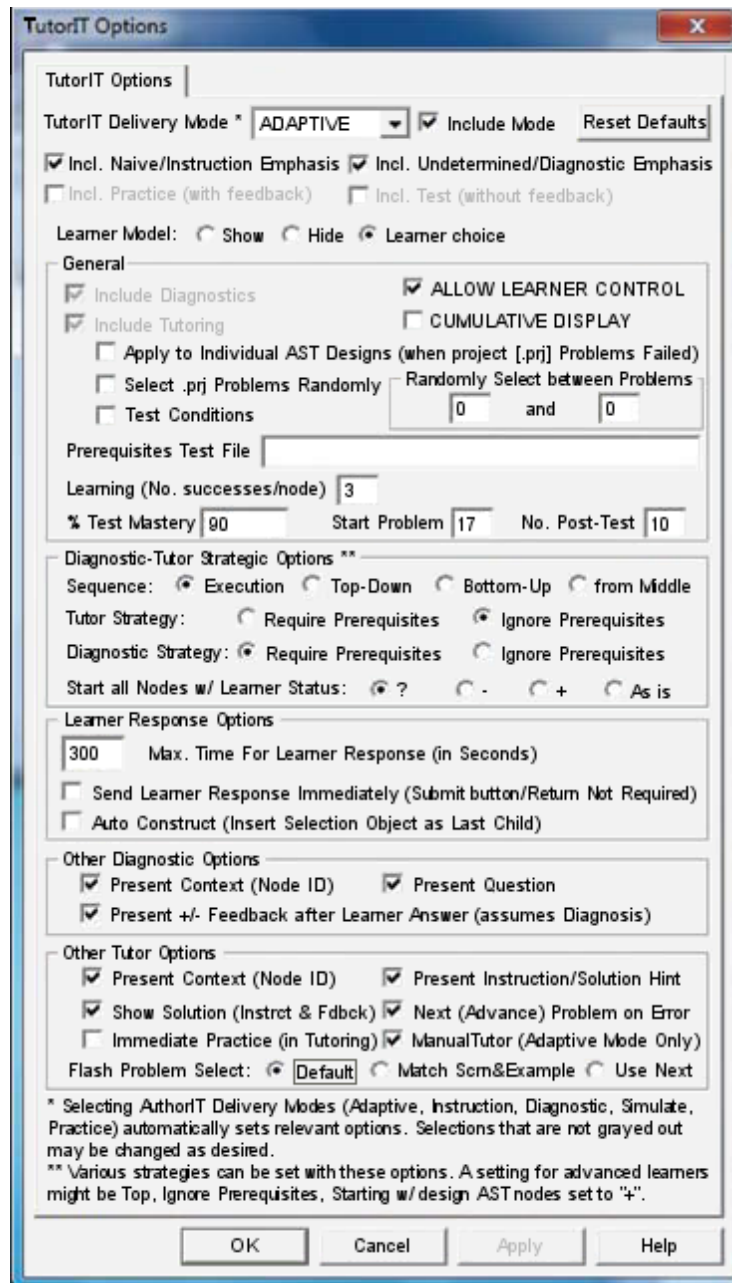


Abbildung 3.1: Das Auswahlwerkzeug des AuthorIT Systems [Scandura 2011].

Leistung aufnimmt. Das TutorIT System kann im Tutor- oder Schülermodus betrieben werden. Die Auswahl ist nach dem Start des Programmes zu treffen, kann jedoch auch später geändert werden. Die Ausführung von TutorIT im Schülermodus ist in Abbildung 3.2 ersichtlich. Sie zeigt die Anwendung während der Umsetzung einer Multiplikationsaufgabe. Als Anmerkung sei die zeilenförmige Darstellungsform des Multiplikationsbeispiels erwähnt, welche das Lernprogramm verwendet. Die Grafik rechts der Berechnung gibt fortlaufend Tipps sowohl grafischer wie akustischer Natur, die zu einer erfolgreichen Durchführung der Aufgabe beitragen sollen. Die Lösungsfindung selbst erfolgt schrittweise. Auf Fehler wird umgehend aufmerksam gemacht, während dazu explizite Hinweismeldungen ausgegeben werden. Eine fehlerfreie Lösung eines Problems beziehungsweise Teilproblems wird ebenso per grafischer Anzeigen mitgeteilt, zusätzlich auch meist mittels anderer Elemente (Animation, Ton, Video, et cetera). Neben der Information über die Korrektheit der Lösung ist derartiges Feedback dafür gedacht, für zusätzliche Motivation bei den Schülern zu sorgen. In der Statusleiste der Anwendung wird außer dem aktuellen Lernfortschritt auch die Anzahl an Problemstellungen mit den prozentual korrekt gelösten Aufgaben dargestellt.

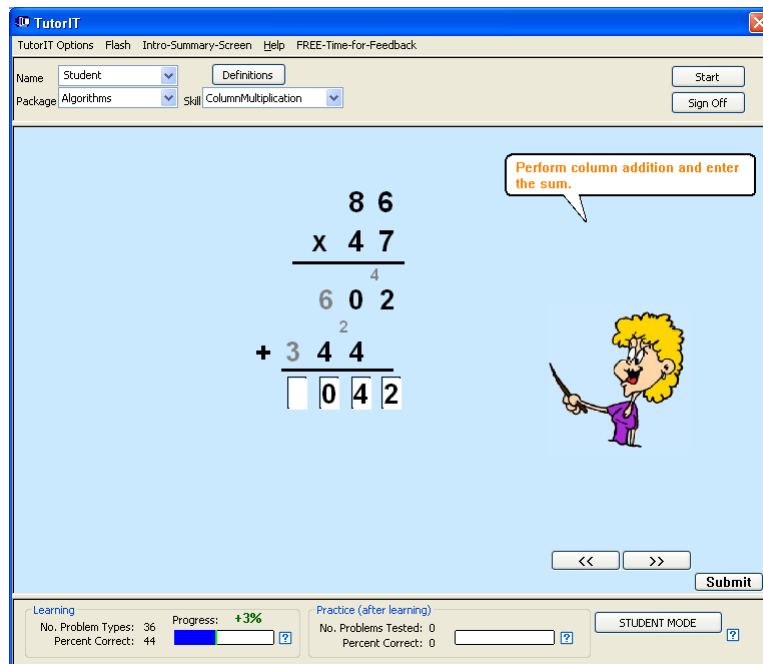


Abbildung 3.2: TutorIT im Schülermodus mit einer Multiplikationsaufgabe [Scandura 2011].

Wird das System im Tutormodus zur Anwendung gebracht, so ist jederzeit der gegenwärtige Lernzustand der Schüler ersichtlich. Dies wird durch die Modellierung der Schülerin beziehungsweise des Schülers realisiert, die sich aus der Verwendung mit untersuchungsrelevanten Flexforms ergibt. Das daraus abgeleitete Lernermodell wird in einer Baumstruktur abgebildet, in der alle erfolgreich gelösten (+), fehlerhaften (-) sowie die noch ungelösten (?) Teilaufgaben mit dementsprechender Kennzeichnung gelistet sind. Wird folglich ein treffendes Problem oder Teilproblem vom Programm ausgewählt, erfolgt intern dessen Lösungsermittlung. Danach wird das Beispiel gemäß der im Blackboard Editor erzeugten Form zur Anzeige gebracht, um von den Schülern gelöst zu werden. Dessen

Antwort wird mit der internen Lösung verglichen. Je nach Ergebnis wird passendes Feedback gegeben. Zusätzlich wird der ausgewählte Zweig im Lernermodell aktualisiert, bei erfolgreicher Lösung mit (+), bei Fehlern mit (-), vorausgesetzt der Anfangszustand der Aufgabe zeigte zuvor ein ungelöstes Problem (?). Wurde ein früheres Beispiel nicht gelöst, markiert durch den entsprechenden Status (-), und erfolgt danach eine korrekte Lösung desselben Problems, so wird die Aufgabe erneut als ungelöst (?) gekennzeichnet. Alle Beispiele mit positiver Markierung (+) gelten als erfolgreich absolviert und werden nicht mehr zur Anzeige gebracht.

Eine bedeutende Eigenschaft von TutorIT nimmt die dynamische Schlussfolgerung aus abgearbeiteten Problemstellungen ein. Das Programm nimmt bei richtigem Ergebnis einer Aufgabe an, dass auch alle ihr untergeordneten Probleme, von denen diese Aufgabe abhängt, *gekonnt* (+) werden, während eine fehlerhafte Lösung alle Probleme, die darauf aufbauen, als *nicht gekonnt* (-) markiert. Die systeminterne hierarchische Anordnung von Wissen ist dafür maßgebend. Versuchen sich, als Beispiel, Schüler erfolgreich an der Lösung von Aufgaben zur mehrstelligen Multiplikation, so wird davon ausgegangen, dass diese Schüler auch deren zugrundeliegende Teilprobleme wie die Addition sowie die einfache Multiplikation ohne Übertrag beherrschen. Im gegenteiligen Fall wird angenommen, dass Schüler etwa Multiplikationsbeispiele mit Übertrag nicht zu lösen imstande sind, sofern bereits Produkte ohne Übertrag Probleme bei der Lösungsfindung bereiten.

In Darstellung 3.3 ist nun die Programmausführung im Tutormodus zu sehen. Im linken Fenster ist die grafische Aufgabenstellung abgebildet, welche auch den Schülern dargelegt wird. Rechts davon ist das Lernermodell in Baumstruktur dargestellt, mit allen entsprechend markierten Zweigen der Teilprobleme. Die Statusleiste im unteren Bereich zeigt wiederum eine Zusammenfassung aller Antworten. Ausbilder haben ferner die Möglichkeit, Einstellungen an der Programmausführung vorzunehmen, wie etwa die Festsetzung eines Zeitlimits zur Lösungsfindung. Dadurch kann schlussendlich die Entscheidungsfindung des Systems, welche sich einzig und allein an der Struktur des Lerninhalts orientiert, beeinflusst werden. Sie ist unabhängig vom Inhalt selbst und verläuft automatisch, was einen immensen Vorteil des Systems ausmacht.

Um sicher zu gehen, dass Lerner für mathematische Problemkategorien, welche in TutorIT durch Tutorials repräsentiert werden, gerüstet sind, setzt das System auf unabdingbare kategoriespezifische Grundvoraussetzungen, die Schüler für erfolgreiches Lernen je nach gewählter Sparte in ebendieser erfüllen müssen. Vorwissen nimmt einen bedeutenden Faktor bei der Lösung eines Tutorials ein. So können abhängig vom Bildungsgrad der Schüler Vorkenntnisse für bestimmte Unterrichtsbereiche oder explizite Problemstellungen vom Programm angenommen werden. Wird als Beispiel das Tutorial über einfache Multiplikationen ausgeführt, setzt das Programm für Benutzer voraus, mit Tastatur und Maus umgehen und Bildschirmanweisungen befolgen zu können. Dies wird aufgrund der in diesem Fall vorherrschenden Trivialität ohne weitere Kontrolluntersuchung von TutorIT als gegeben hingenommen. Handelt es sich aber um umfassendere Problemkategorien wie die mehrstellige Multiplikation, so finden sehr wohl Überprüfungen von Vorkenntnissen dazu statt. Dieser Prozess soll das Wissen über Teilprobleme, aus welchen sich die eigentlichen Aufgaben dieser Rubrik zusammensetzen, gewährleisten und letztendlich Lernerfolg sicherstellen. Im Fall der mehrstelligen Multiplikation müssen zu Beginn des Tutorials Aufgabenstellungen über einfache Multiplikationen und Additionen gelöst werden, die Grundvoraussetzung für den betreffenden Bereich sind. Erst nach erfolgreicher Absolvierung dieser Art Einstiegstests stehen die tatsächlichen Aufgaben, hier mehrstellige Multiplikationen, zur Lösungsfindung bereit. Arbeiten sich Schüler mit entsprechenden Vorkenntnissen erfolgreich durch die gesamten Aufgaben eines Tutorials, so kann mit Sicherheit davon ausgegangen



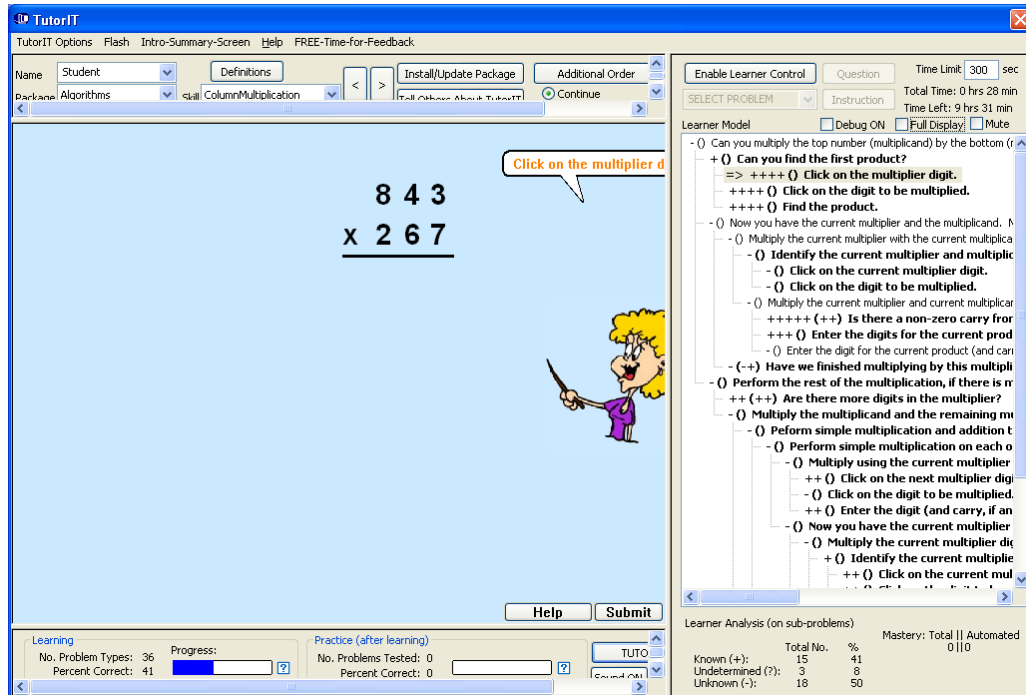


Abbildung 3.3: Ausführung von TutorIT im Tutormodus [Scandura 2011].

werden, dass diese das gewählte Fachgebiet des Tutorials und somit das Können dessen beherrschen. Diese Vorgehensweise ist sinnvoll, da sich beim Lösen komplexer Aufgaben von Schülern ohne Basiswissen ein Lernerfolg nur mäßig, wenn überhaupt, einstellt. Eher werden diese dazu geneigt sein, das Programm aufgrund von sinkender Motivation vorzeitig zu beenden. Daher ist es ebenso Aufgabe des Systems, neben der bestmöglichen Unterstützung für eine erfolgreiche Lösungsfindung auch Motivation und Ansporn für die vollständige Abarbeitung eines Tutorials zu liefern, um dadurch Lernen garantieren zu können. Die Dauer des Lernvorgangs spielt dabei eine untergeordnete Rolle und hängt einzig und allein von der Schülerin oder vom Schüler selbst ab.

Das vorgestellte Lernsystem hat keinesfalls zum Ziel, menschliche Ausbilder in den Hintergrund zu rücken oder diese gar vollkommen zu ersetzen. Vielmehr ist dem Autor bewusst, dass sowohl computergestützte Systeme als auch Lehrende für erfolgreiches Lernen entscheidend sind und von gemeinsamen Synergien profitieren können. Ein maximaler Lernerfolg kann nur durch einen gemeinsamen pädagogischen Einsatz zur vollen Ausprägung kommen.



## 4 Technische Umsetzung des Prototyps

Dieser Abschnitt behandelt die Entwicklung und technische Umsetzung der mathematischen Lernanwendung. Diese nennt sich *Mathe-Multi-Trainer*, in Anlehnung an dessen Funktion als Lernprogramm für mehrstellige Multiplikationen. Der Name wurde bewusst einfach und für Schüler leicht merkbar gewählt. Dies ist für den Einsatz in Volks- und Mittelschulen eine nicht unwesentliche Anforderung. Das Programm wurde als Web-Applikation entwickelt und liegt auf einem für mathematische Lernapplikationen ausgelegten Apache-Webserver<sup>1</sup> der Technischen Universität Graz – der sogenannten Mathematischen Lernplattform<sup>2</sup>. Ein Direktlink zum entwickelten und hier vorgestellten Mathe-Multi-Trainer ist als Fußnote<sup>3</sup> gegeben. Es sei angemerkt, dass vor der Benutzung dieser und weiterer auf der Plattform verfügbaren Anwendungen eine Benutzerregistrierung über ebendiese Plattform vonnöten ist, um Lehrpersonen und Schüler getrennt voneinander behandeln zu können und exakte Klassenauswertungen sicherzustellen.

Der Prototyp des Mathe-Multi-Trainers kann mit allen gängigen Browsern ausgeführt werden. Eine permanente Verfügbarkeit des Lernprogrammes für dessen Anwender ist somit gegeben. Durch die Ausbreitung des Internets und dessen alltäglichen Einsatz kann außerdem eine gewisse Erfahrung und Vertrautheit mit Internetanwendungen auch bei Volks- und Mittelschülern vorausgesetzt werden. Abgesehen von dieser Tatsache sollten Schüler trotzdem gewissenhaft in das Programm eingeführt werden, vorzugsweise von vertrauten Lehrkräften. Die Benutzerführung der Anwendung ist diesbezüglich einfach und volks- beziehungsweise mittelschulgerecht gehalten.

Nachfolgend werden die Grundlagen und verwendeten Technologien der Lernapplikation erklärt, ehe die Funktionsweise und erwähnenswerte Programmausschnitte vorgestellt werden. Ein wichtiger Teil lenkt den Fokus auf die entwickelten Algorithmen zur Generierung der mehrstelligen Multiplikationen und zur Auswertung und Fehleranalyse der Benutzereingaben. Das Datenbankdesign des Prototyps wird erst nach diesen Beschreibungen zu Beispielgenerierung und Ergebnisevaluierung präsentiert, da Vorwissen zu Struktur und Attributen der Datenbank und dessen Tabellen in ebendiesen Abschnitten vermittelt wird.

### 4.1 Verwendete Technologien

#### 4.1.1 PHP 5

Die Implementation des Mathe-Multi-Trainers basiert zur Gänze auf der Skriptsprache PHP (Hypertext Preprocessor)<sup>4</sup>. Die Entscheidungsgrundlagen zur Auswahl von PHP bildeten die Popularität der

<sup>1</sup> <http://httpd.apache.org/> [Zugriff am 21.08.2012]

<sup>2</sup> <http://mathe.tugraz.at/> [Zugriff am 21.08.2012]

<sup>3</sup> <http://mathe.tugraz.at/~mickname/> [Zugriff am 21.08.2012]

<sup>4</sup> <http://www.php.net/> [Zugriff am 21.08.2012]

Sprache, dessen Spezialisierung auf webbasierte Entwicklung – hier insbesondere dessen dynamische Eigenschaften – und dessen anwenderfreundliche Interaktionsmöglichkeiten mit und Einbettung in HTML-Darstellungsseiten. Die Anwendung beruht auf PHP 5, der Version 5 der Sprache. Änderungen, Verbesserungen und Bug-Fixes zu früheren Versionen von PHP sind auf dessen offizieller Homepage (siehe oben) zu finden. Durch eine Open-Source-Software-Lizenz, der PHP-Lizenz<sup>5</sup>, ist die freie Verwendung der Sprache beziehungsweise Verbreitung von auf PHP basierendem Sourcecode sichergestellt. PHP ist eine typschwache serverseitige Programmiersprache, das heißt PHP-Skripts werden auf einem Webserver ausgeführt und verarbeitet. Dadurch können Webseiten in dynamischer Weise dargestellt werden. Die Funktionsweise lässt sich an folgendem allgemeinen Beispiel demonstrieren. Eine Schülerin oder ein Schüler ruft beispielsweise die vorgestellte Lernapplikation durch Eingabe der URL in einem Browser auf. Danach läuft die Applikation und die Anwender bestimmen durch ihre Eingaben den weiteren Programmablauf. Durch diese Interaktion mit dem Programm werden laufend Anfragen für PHP-Seiten an den Server gestellt, sprich gesendet. Vereinfacht gesagt sieht sich der Server ein PHP-File mit den übertragenen individuellen Daten der Benutzer an, ein sogenannter PHP-Interpreter erzeugt anhand dieser Daten Code für die Ausgangsdatei, die daraufhin vom Webserver wieder an den Aufrufenden zurückgesendet wird. Die Benutzer sehen nun im Browser die Reaktion der Applikation auf ihre vorhergehenden Aktionen (Eingaben, Mausclicks, et cetera). [Schütze und Graen 2006]

#### 4.1.2 MySQL

Eng verknüpft mit der Verwendung von PHP ist in diesem Zusammenhang das relationale Datenbanksystem MySQL<sup>6</sup>, welches als Datenbankverwaltungssystem des Mathe-Multi-Trainers, also als Datenbankanbindung der Applikation, eingesetzt wird. Die Verwendung von MySQL geht einher mit der Erstellung von dynamischen Webseiten durch PHP. MySQL Datenbanken stehen unter der Open-Source-Lizenz und sind in deren Anwendung großflächig verbreitet. Durch ein webbasiertes Datenbank-Back-End ist zudem eine einfache Konfiguration der Datenbanken möglich. Diese Schnittstelle nennt sich phpMyAdmin und basiert – wie der Name schon sagt – auf PHP und ermöglicht die Bearbeitung und Verwaltung von Datenbanken, Tabellen und Daten, welche mit der Applikation verknüpft sind. Durch die ausgezeichnete Zusammenarbeit von PHP und MySQL gewährleistet deren gemeinsame Verwendung eine starke und stabile Applikation. [Oracle 2012]

#### 4.1.3 CSS

Zur Gestaltung und Formatierung der erzeugten HTML-Seiten im Front-End – also im Browser des Benutzers – werden Cascading Style Sheets (CSS)<sup>7</sup> eingesetzt. Dadurch ist sichergestellt, dass Inhalt und Design in optimaler Weise getrennt voneinander behandelt werden können. Diese Vorgehensweise trägt wesentlich zu einem gut strukturierten Aufbau der Implementation und zur Entwicklung von wiederverwendbarem Code bei. Zusammen mit HTML<sup>8</sup>, Kurzform für Hypertext Markup Language und als Auszeichnungssprache somit verantwortlich für den Aufbau und die eigentliche Struktur der Inhalte, bilden CSS das grundlegende Erscheinungsbild des Mathe-Multi-Trainers. Der Prototyp

<sup>5</sup> [http://www.php.net/license/3\\_01.txt](http://www.php.net/license/3_01.txt) [Zugriff am 21.08.2012]

<sup>6</sup> <http://www.mysql.com/> [Zugriff am 21.08.2012]

<sup>7</sup> <http://www.w3.org/Style/CSS/> [Zugriff am 21.08.2012]

<sup>8</sup> <http://www.w3.org/html/> [Zugriff am 21.08.2012]

des Lerntrainers verwendet Cascading Style Sheets diesbezüglich für alle Designfragen (Layout, Ausrichtung der Eingabezellen, et cetera). In naher Zukunft ist geplant, das Design von Experten an ein professionelles Volks- beziehungsweise Mittelschullayout anzupassen. Dies sollte durch die vorliegende strikte Trennung von Inhalt und Gestaltung in der Implementation ohne Probleme möglich sein.

#### 4.1.4 jQuery

Vereinzelt wurde bei der Entwicklung der Lernanwendung auf jQuery<sup>9</sup> zurückgegriffen, diese Technologie wurde daher in die Applikation eingebunden. Bei jQuery handelt es sich um ein Open-Source-Framework, genauer gesagt um eine JavaScript-Bibliothek, welche dynamische Funktionen zur Navigation in Dokumenten, zum Event-Handling und zur Manipulation und Selektion von Elementen im Document Object Model (DOM) von HTML-Seiten zur Verfügung stellt. Im Mathe-Multi-Trainer wurden jQuery-Funktionen hauptsächlich bei der Bearbeitung und Darstellung der Zellen zur Eingabe der Multiplikationsantworten verwendet. [jQuery Foundation 2012, Weyl 2008]

#### 4.1.5 Zend Framework

Das Fundament der Implementation des Prototyps stellt das Zend Framework<sup>10</sup> in der Version 1.11.10 dar. Wie alle anderen verwendeten Technologien ist es als Open-Source-Software frei verfügbar und der Quellcode frei veränder- und verteilbar. Das Zend Framework ist in der Programmiersprache PHP – genauer in PHP 5 – geschrieben. Da es auf die Entwicklung von Web-Anwendungen spezialisiert ist und darüber hinaus einfach zu verwenden ist, wurde es als Grundlage bei der Implementierung des Mathe-Multi-Trainers eingesetzt. Weitere Vorteile des Frameworks bieten dessen einfache Erweiterbarkeit, dessen ausgiebig geprüfter und stabiler Code, sowie die volle Objektorientiertheit der gesamten Funktionalität. Der Aufbau des Frameworks stützt sich auf eine Vielzahl von Komponenten, welche individuell oder gemeinsam eingesetzt werden können. Die generelle Architektur des Zend Frameworks bietet einen hohen Grad an Flexibilität bei der Anwendung und führt in der Regel zu einer Verkürzung der Programmierzeit im Vergleich zu *reinen* PHP-Anwendungen. [Zend 2012a]

Eine mächtige Komponente des Frameworks stellt der sogenannte Model-View-Controller (MVC) dar. Bei Verwendung schafft dieser die Grundstruktur eines wohldefinierten Designs einer Webapplikation. Das Prinzip des Model-View-Controllers ist es, den Code für Präsentation, Geschäftslogik und Datenzugriff strikt getrennt zu halten, um damit eine Gliederung des Quellcodes in die drei genannten Bereiche zu erreichen, den Code übersichtlicher zu gestalten und in weiterer Sicht die Entwicklung zu simplifizieren. Die View-Komponente ist hauptsächlich dafür zuständig, Daten an die Benutzer weiterzuleiten und darzustellen, und Eingaben entgegenzunehmen. Der Datenzugriff und die grundlegende Funktionalität geschieht über die Model-Komponente, während der Controller für die sorgfältige Zusammenarbeit aller drei Teile, die korrekte Auswahl von Views zur Anzeige, wie auch für die exakte Weiterleitung der Daten an die jeweiligen Komponenten oder Routinen der Model-Instanzen zuständig ist. Die Implementation des Prototypen umfasst einen Model-View-Controller mit oben erwähnten Eigenschaften. [Zend 2012b]

<sup>9</sup> <http://jquery.com/> [Zugriff am 21.08.2012]

<sup>10</sup> <http://framework.zend.com/> [Zugriff am 21.08.2012]

## 4.2 Konzept und Funktionalität

Die Entwicklung eines mathematischen Lernprogrammes für Schüler ab der vierten Leistungsstufe zum Erlernen, Üben und Festigen der mehrstelligen Multiplikation bedarf einer sensiblen Konzepterstellung, in der speziell die Eigenschaften und Eigenheiten der Benutzerzielgruppe – also Kinder ab einem Alter von circa zehn Jahren – in Betracht gezogen werden müssen. Dabei spielt einerseits ein kindergerechtes Layout der grafischen Benutzerschnittstelle der Applikation eine gewichtige Rolle, andererseits ist eine einfache und *spielende*, aber auch zielgerichtete Handhabung und Benutzerführung der Applikation von enormer Bedeutung. Eine einwandfreie Funktionalität wird vorausgesetzt.

Schüler sollen Spaß und Motivation beim Üben mit dem Programm entwickeln. Daher werden anfangs für jede Schülerin beziehungsweise für jeden Schüler Multiplikationsaufgaben aus der einfachsten Leistungskategorie des Beispiel-Definitions-bereichs des Mathe-Multi-Trainers – im Programm als sogenannte *Kompetenz* gekennzeichnet – zur Anzeige gebracht (genauer dazu weiter unten und in Unterkapitel 4.4). Es würde keinen Sinn ergeben, Schüler unmittelbar nach Übungsbeginn mit komplexen Beispielen zu überfordern, da deren Leistungsfähigkeit dem Programm noch nicht bekannt ist und Benutzer vor unlösbaren Problemen stehen könnten. Das würde zu einer negativen Grundeinstellung gegenüber dem Lerntrainer führen, was vermieden werden soll. Die Applikation soll behutsam auf die Leistung der Schüler eingehen und je nach individuellen Stärken oder Schwächen leistungsgerechte Beispiele erzeugen. Das Programm soll sich in adaptiver Weise den Leistungen der Schüler anpassen. Als Darstellungsform für die Multiplikationen in der Lernanwendung wird die sogenannte Schulmethode der Mathematik verwendet, wie in Beispielrechnung (3.2) aus Kapitel 3 zu sehen.

Wie im letzten Absatz angemerkt, soll die Lernanwendung verschiedene Beispielkategorien abdecken, die sich jeweils in ihrer Schwierigkeit unterscheiden. Der Definitionsbereich einer Aufgabenkategorie soll dabei durch die Stellenanzahl von Multiplikand und Multiplikator eingegrenzt werden. Die Ermittlung von Multiplikationen für eine bestimmte Problemgruppe soll dann immer mit zufällig generierten Zahlen erfolgen, welche die Spezifikation für eine gewählte Aufgabenkategorie erfüllen müssen. Diese Vorgehensweise soll für alle Problemgruppen eine vielfältige und universelle Beispielerzeugung sicherstellen, in der keine Vorhersagen über die nächste Aufgabe getroffen werden können und eventuelle kognitive und empirische Manipulationsversuche wie Auswendiglernen von aufeinanderfolgenden Problemstellungen, als Beispiel, automatisch im Keim erstickt werden.

Grundsätzlich ist der allgemeine Definitionsbereich des Mathe-Multi-Trainers für mehrstellige Multiplikationen auf Multiplikanden mit zwei bis vier Stellen und auf Multiplikatoren mit ein bis drei Stellen festgesetzt. Es sollen keine Beispiele generiert werden, deren Multiplikand eine größere Stellenanzahl aufweist als sein entsprechender Multiplikator. Dadurch ergeben sich acht verschiedene Problemgruppen, welche in Tabelle 4.1 gelistet sind. Die Variable  $n$  stellt hier den Ziffernwert einer einstelligen Zahl dar und kann Werte zwischen 0 und 9 einnehmen.

Die acht Aufgabengruppen werden in weiterer Folge in drei Kernkompetenzen zusammengefasst. Ausschlaggebend dafür ist die Stellenanzahl des Multiplikators. Wie in Tabelle 4.1 zu sehen, ergeben Aufgaben mit einem einstelligen Multiplikator, also Problemgruppe 1 bis Gruppe 3, die Kompetenz I. Sie markiert gleichzeitig die einfachste Leistungsstufe. Multiplikationen mit zweistelligen Multiplikatoren (Gruppe 4 bis Gruppe 6) werden in Kompetenzkategorie II zusammengefasst. Aufgaben mit dreistelligen Multiplikatoren (Gruppe 7 und Gruppe 8) ergeben sich zu Kompetenz III, die gleichbedeutend die komplexeste und somit schwierigste Leistungskategorie darstellt.

Kompetenz	Gruppe	Multiplikationsmuster
I	1	$nn * n$
	2	$nnn * n$
	3	$nnnn * n$
II	4	$nn * nn$
	5	$nnn * nn$
	6	$nnnn * nn$
III	7	$nnn * nnn$
	8	$nnnn * nnn$

Tabelle 4.1: Die drei Kompetenzstufen mit den acht Problemgruppen und den entsprechenden Multiplikationsmustern des Mathe-Multi-Trainers.

Somit umfasst der Mathe-Multi-Trainer drei weitreichende Leistungsstufen, die Kompetenzen. Diese stellen für Schüler in aufsteigender Reihenfolge eine erhebliche Zunahme in Komplexität und Schwierigkeit dar, bieten folglich eine ausgezeichnete Möglichkeit zum schrittweisen Erlernen und Üben der mehrstelligen Multiplikation. Den größten Sprung bei der Komplexitätssteigerung markiert der Übergang von Kompetenz I zu Kompetenz II, da Beispielaufgaben mit zweistelligen Multiplikatoren erstmals eine Gesamtaddition von zuvor errechneten Teilprodukten vorgeben. Dies ist in Kompetenzstufe I nicht der Fall, da einstellige Multiplikatoren in einer einzigen Ergebniszeile gelöst werden. Dass sich die einer Kompetenzkategorie untergeordneten Problemgruppen ebenso in deren Schwierigkeiten voneinander unterscheiden, ist Ergebnis der variablen Stellenanzahl des Multiplikanden, des linken Faktors der Multiplikationsrechnung. Im Gegensatz zur Stellenzunahme des Multiplikators ist die Komplexitätssteigerung von entsprechenden Multiplikationen innerhalb einer Kompetenzkategorie, beispielsweise von Gruppe 2 auf Gruppe 3, nicht groß genug, um daraus eine eigene Kompetenzstufe zu bestimmen. Die geringfügige Schwierigkeitsschwankung liegt in einer (zum Beispiel von Gruppe 1 auf Gruppe 2) beziehungsweise zwei zusätzlichen einfachen Multiplikationen (zum Beispiel von Gruppe 1 auf Gruppe 3), welche Benutzer bei Stellenzunahme des Multiplikanden zu lösen haben. Eine Anhebung der Stellenanzahl des Multiplikators hat dagegen immer mindestens eine Verdoppelung des gesamten Rechenaufwandes für eine Problemstellung zur Folge (zum Beispiel von Kompetenz I auf Kompetenz II). Die verschiedenen Gruppen einer Kompetenzkategorie vergrößern somit die Menge an möglichen Problemstellungen für eine Leistungsstufe in sehr hohem Maße und machen die Aufgabenermittlung damit äußerst breitgefächert und noch umfangreicher.

Eine weiteres signifikantes Unterscheidungskriterium in der Schwierigkeitseinstufung von Multiplikationen sind Aufgabenstellungen, welche ohne Übertrag lösbar sind, und Multiplikationsrechnungen, welche einen Übertrag in ihrer Lösung miteinbeziehen. Ausschlaggebend dafür, um welche der beiden Arten es sich bei einer Multiplikation handelt, sind einzig und allein die verwendeten Ziffern und Zahlen sowohl des Multiplikanden als auch des Multiplikators der Multiplikation. Diese Überlegung führt zu einer zusätzlichen übergeordneten Unterteilung der Leistungsstufen und Schwierigkeitseinteilung der Aufgabenstellungen des Mathe-Multi-Trainers. Multiplikationen mit Übertrag sind generell schwieriger einzustufen als jene ohne Übertrag, da für die positive Lösungsfindung

von Übertragsbeispielen zusätzliches Wissen wie eben das Aufaddieren von Überträgen auf das nachfolgende Teilprodukt notwendig ist. Der Definitionsbereich für Beispiele ohne Übertrag und Beispiele mit Übertrag bleibt prinzipiell derselbe wie in Tabelle 4.1 dargestellt. Der Unterschied beider Leistungsstufen liegt in der Vorgabe an die Algorithmen zur Problemgenerierung, passende Zahlen für Multiplizanden und Multiplikator derart zu ermitteln, dass entweder einfache Multiplikationen ohne Übertrag oder eben Multiplikationen mit Übertrag erzeugt werden.

Zusammengefasst ergeben sich somit für die Aufgabenstellungen des Mathe-Multi-Trainers zwei übergeordnete getrennte Kompetenzstufen: *MIT* und *OHNE* Übertrag. Beide Klassen gliedern sich in jeweils drei weitere Kompetenzkategorien auf, welche wiederum bis zu drei unterschiedliche Problemgruppen beinhalten. Insgesamt sind also 16 verschiedene Modi für Aufgabenstellungen zu berücksichtigen.

Treten bei der Beantwortung von Aufgabenstellungen aus einer bestimmten Kompetenzkategorie kontinuierlich Lösungsschwierigkeiten auf, so führt das bei Schülern unweigerlich zu einer Motivationsreduktion. In diesem Fall ist die verantwortliche Lehrperson dafür zuständig, passend in den Lern- beziehungsweise Übungsprozess der betroffenen Schüler einzugreifen und Gegenmaßnahmen zu setzen. Angesprochene Lehrkräfte haben diesbezüglich auf eine umfassende Statistikauswertung Zugriff. Diese ist nur für Lehrpersonen sichtbar und stellt alle vom Programm generierten Aufgaben einschließlich der Lösungseingaben pro Schülerin oder Schüler zur Verfügung. Fehleralgorithmen werden auf die einzelnen Antworteingaben der Problemstellungen angewandt und bestimmen den Grund für allfällige Schwierigkeiten bei der Lösungsfindung. Genaue Angaben zu Fehlern in der Statistikauswertung zielen auf ein schnelles und unmittelbares Eingreifen der Lehrkräfte in den Lernprozess der Schüler ab. Ohne diese Eigenschaft des Mathe-Multi-Trainers würden Lehrpersonen ein beträchtliches Ausmaß an wertvoller Zeit bei der Suche und Ermittlung von Fehlern oder einer allgemeinen Fehlerhäufung in der Aufgabenmenge der Schüler verlieren. Die automatisierte Fehlerbestimmung der Lernapplikation wird in Abschnitt 4.5, die Statistikauswertung in Abschnitt 4.7 erklärt.

#### 4.2.1 Allgemeiner Aufbau der Implementation

Wie in Unterkapitel 4.1.5 angemerkt, basiert die Umsetzung des Mathe-Multi-Trainer Prototyps auf dem Model-View-Controller Architekturmuster. Dadurch ist nicht nur eine modulare Trennung von Logik und Datenhaltung, Präsentation und Programmsteuerung gegeben, dieses Feature trägt auch wesentlich zu einem wiederverwendbaren und erweiterbaren Quellcode bei. Die Implementierung enthält drei essenzielle Controller (neben weiteren), welche maßgeblichen Einfluss auf den Programmfluss ausüben. Diese sind nachfolgend aufgelistet, ehe deren Funktionalität im Anschluss daran erklärt wird:

- AccessController
- CalculationController
- StatisticController

**AccessController:** Diese Klasse ist namensgebend für den An- und Abmeldemodus der Applikation. Sie nimmt Benutzernamen und Passwort der Anwender vom zentralen Login-Formular der Eingangs-



seite entgegen und prüft die eingegebenen Daten auf deren Authentizität. Über ein Plugin wird dabei via Webservices mit dem Usermanager des Servers kommuniziert, der für die allgemeine Benutzerverwaltung verantwortlich ist und damit auch die Benutzerüberprüfung durchführt. Das Passwort wird durch einen sicheren Hash-Algorithmus (SHA-256) verschlüsselt übertragen. Der Usermanager ist es auch, der Benutzern eine Rolle zuteilt, in diesem Fall eine der Rollen Schüler, Lehrperson oder Administrator. Dadurch können bei der Benutzung der Applikation diverse Privilegien oder Einschränkungen für User mit einer bestimmten Rolle angewendet werden. Neben dem Login-Vorgang ist der AccessController ebenso für die Abmeldung der Benutzer vom Programm zuständig.

**CalculationController:** Die Hauptaufgabe des CalculationControllers liegt in der Steuerung der Beispielgenerierung und des Evaluierungsvorgangs der Benutzerantworten. Durch Interaktion mit der Model-Komponente erzeugt die Anwendung eine für den eingeloggtten Benutzer passende Aufgabenstellung und schickt die erzeugte Multiplikation zur Anzeige (View), gemeinsam mit einem für die Rechnung entsprechend ausgelegten Eingabeformular. Nach Eingabe der Ergebniszellen und Klick auf den Antwort-Button durch den Benutzer wird je nach Problemgruppe der Multiplikation eine passende Instanz einer Auswertungsmethode der Model-Komponente aufgerufen und die Benutzerantworten von zuvor als Parameter zur Evaluierung mitübergeben. Die Korrektheit oder falsche Lösung der untersuchten Multiplikationsrechnung wird dem Anwender sogleich grafisch und in schriftlicher Form am Bildschirm mitgeteilt.

**StatisticController:** Dieser Controller verantwortet und steuert die Statistikauswertung für Lehrkräfte. Die Aufstellungen beruhen sich dabei auf die gespeicherten Antworten der Benutzer zu Multiplikationsaufgaben. Eine Lehrkraft hat nur Zugriff auf die Auswertungen der ihr zugewiesenen Klasse oder Klassen, genauer gesagt der Schülerinnen und Schüler dieser Klassen – die Zuweisung ist Aufgabe des Usermanagers und wird von externer Stelle aus verantwortet. Die verschiedenen Statistiken können sich auf generelle Zusammenfassungen und Feststellungen aller beantworteten Multiplikationen eines Benutzers belaufen, es können aber auch entsprechende Verfeinerungen bis zur Darstellung jeder einzelnen Benutzereingabe mitsamt Fehlerstatus ausgewählt werden. In der Gesamtheit betrachtet können absolut alle einzelnen Userantworten in der Statistikauswertung nachvollzogen werden.

Ein weiterer erwähnenswerter Controller der Implementation ist der SoapController. Dieser regelt das Handling von angebotenen Webservices des Mathe-Multi-Trainers, welche über das SOAP-Protokoll<sup>11</sup> übertragen werden. Mehr dazu in Abschnitt 4.9.

Die zwei wichtigsten Funktionalitäten der vorgestellten Lernanwendung sind die entwickelten Algorithmen zur Erzeugung der Problemstellungen und zur Auswertung der Benutzereingaben. Beide werden in Abschnitt 4.4 (Generierung) und 4.5 (Evaluierung) im Detail erklärt.

Der Prototyp des Mathe-Multi-Trainers hat die Fähigkeit, zweisprachig ausgeführt zu werden. Die Standardsprache der Applikation ist Deutsch. Auf dessen Einstiegs- beziehungsweise Anmeldeseite ist es möglich, die Anzeigesprache des Programms auf Englisch zu setzen. Dies ist durch einen Link kenntlich gemacht. Die modulare Implementationsweise der Anwendung macht es problemlos und in sehr einfacher Weise möglich, weitere Sprachen verfügbar zu machen und die Applikation dadurch noch polyglotter und flexibler einsetzbar zu machen.

<sup>11</sup> <http://www.w3.org/TR/soap/> [Zugriff am 21.08.2012]

### 4.3 Layout der Benutzeroberfläche

Die Gestaltung der Benutzeroberfläche beruht auf einem Basis-Layout, auf welchem alle generierten View-Skripte aufsetzen. Der Entwurf dieses Fundaments ist für den Prototyp bewusst minimalistisch und einfach gehalten, da sich in absehbarer Zukunft Designspezialisten um das professionelle Aussehen der Lernapplikation kümmern werden. Trotzdem wurde bereits in der Implementierung versucht, auf bestimmte volks- und mittelschulgerechte Merkmale Rücksicht zu nehmen, um erste Tests mit Schülerinnen und Schülern durchführen zu können. Das Layout ist in angenehmen Farben und Schriftarten gehalten, Antwortbuttons ebenso. Die Benutzerführung durch das Programm ist intuitiv und für Kinder leicht verständlich. Abbildung 4.1 zeigt die Anmeldeseite, welche gleichzeitig die Eingangsseite der Applikation präsentiert. Nach Abmeldung vom Lerntrainer – das heißt nach Klick auf den Link *Logout* im Navigationsfenster (nur im eingeloggten Zustand sichtbar) – wird gleichermaßen auf diese Seite verwiesen.



Abbildung 4.1: Die Eingangsseite des Mathe-Multi-Trainers.

Auch die Eintragung der Lösung für die Multiplikationsaufgaben in dafür konzipierte Eingabezellen ist schülergerecht gehalten und ebenso einfach nachzuvollziehen. Im Beantwortungsmodus liegt der Fokus des Cursors immer in der ersten Eingabezelle, welche für eine korrekte Lösung nach der Schulmethode vorgesehen ist. Der Cursorfokus kann durch Betätigung der Tabulatortaste in die nächste vorgesehene Eingabezelle geleitet werden. Es ist auch möglich, den Cursor mit der Maus in eine beliebige Zelle zu setzen. In Abbildung 4.2 ist der Beantwortungsmodus des Mathe-Multi-Trainers dargestellt, ergo die Ausführung des Programms während des Lösungsvorgangs einer Multiplikationsaufgabe. Es ist eine Problemstellung mit Übertrag der einfachsten Kompetenzklasse und Gruppe (siehe dazu Tabelle 4.1) zu sehen, deren erstes Teilprodukt bereits korrekt in die dafür vorgesehenen Eingabezellen eingetragen wurde. Wie im Beispiel ersichtlich, liegt der Fokus des Cursors in der nächsten auszufüllenden Antwortzelle, der Zehnerstelle des Ergebnisprodukts. Nach der vollständigen Lösung der Multiplikation beziehungsweise nach dessen Lösungsversuch mitsamt

Befüllung der entsprechenden Eingabezellen muss rechts daneben der Antwortbutton zum Senden der Gesamtlösung gedrückt werden.

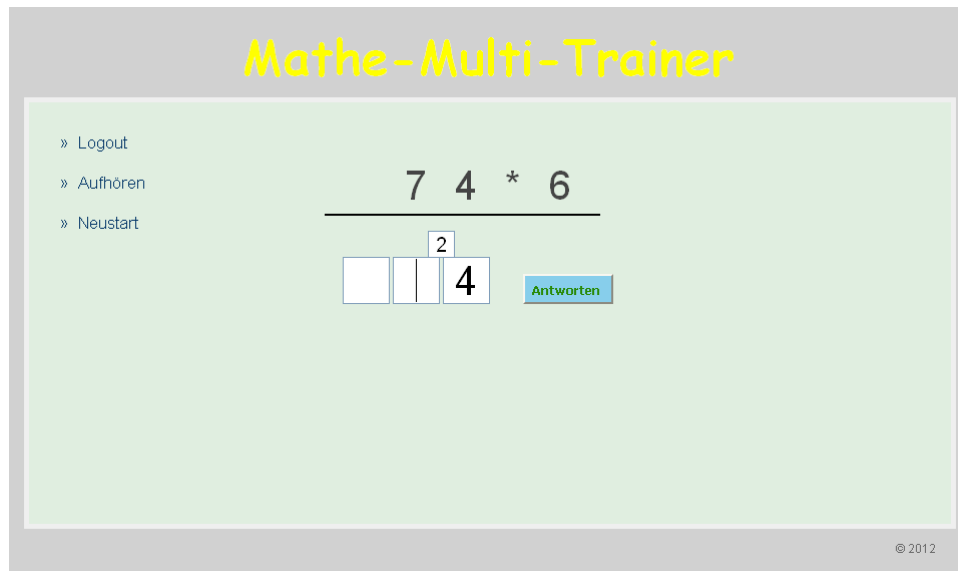


Abbildung 4.2: Die Lernanwendung während der Beantwortung einer Multiplikationsaufgabe.

Der korrekten Lösung einer Aufgabenstellung folgt die Textfolge *Richtiges Endergebnis!* in blauer Farbe. Dazu wird die gerade bewältigte Multiplikationsrechnung einschließlich der vom Benutzer befüllten Ergebniszellen nochmals angezeigt, wobei die Antwortzellen nicht mehr veränderbar sind und ausgegraut (der Opera-Browser stellt hier eine Ausnahme dar) dargestellt werden. Abbildung 4.3 zeigt den Status des Lernprogramms nach Eingabe der korrekten Lösung einer übertragslosen Multiplikation mit vierstelligem Multiplikanden und zweistelligem Multiplikator. Wie beschrieben, wird die fehlerlose Lösung der Anwenderin beziehungsweise des Anwenders textlich und farblich hervorgehoben. Nach Betätigung des Buttons *weiter* wechselt das Programm wieder in den Beantwortungsmodus.

Inkorrekte Ergebniszellen erhalten nach Beantwortung der Problemstellung eine farbliche Kennzeichnung, indem die Zellen eine rote Umrandung erhalten. Dadurch kann auf Fehler unmittelbar aufmerksam gemacht werden. Zusätzlich wird der Text *Leider falsch!* in rotem Farbton angezeigt. In Abbildung 4.4 ist die Anzeige der fehlerbehafteten Beantwortung einer Multiplikation mit Übertrag, bestehend aus einem Multiplikanden mit vier Stellen und einem Multiplikator mit drei Stellen, dargestellt. Auf einen Blick sind alle inkorrekten Eingaben ersichtlich. Es sei angemerkt, dass nicht ausgefüllte Übertragszellen immer als falsch gekennzeichnet sind, obwohl deren Nichtbefüllung zu einer korrekten Lösung führen kann. Dies ist der Fall, wenn sich Benutzer anfallende Überträge von Teilmultiplikationen ohne Eintragung merken und danach im Kopf zum nächsten Teilergebnis erfolgreich addieren. Die Anzeige als Fehler hat den Sinn, Schüler auf die korrekte Lernweise der Übertragungseintragung hinzuweisen. Ferner führt diese Denkhilfe unweigerlich zu einer Fehlerminimierung der Schüler.

Das Layout der Nutzerstatistik basiert auf einfachen Tabellen, in welchen eine Menge nützlicher Anwenderdaten angezeigt werden können. Anfangs erfolgt bei Auswahl des Links *Statistiken* eine



Abbildung 4.3: Die korrekte Lösung einer Multiplikationsaufgabe ohne Übertrag der Kompetenz II und Gruppe 6.

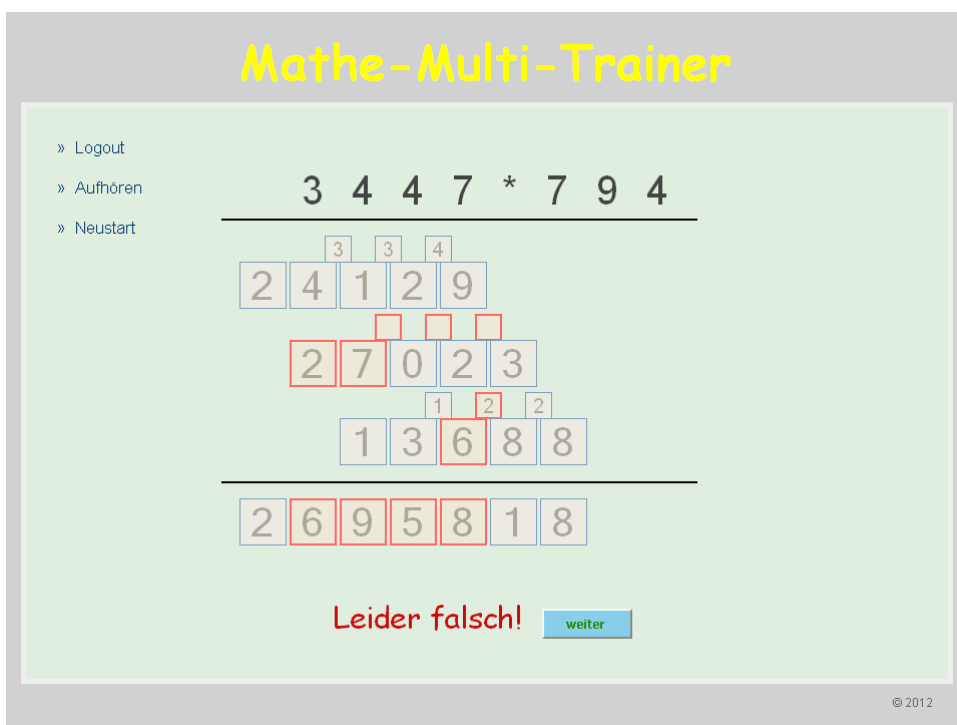


Abbildung 4.4: Der Mathe-Multi-Trainer nach fehlerhafter Lösung einer Multiplikationsaufgabe mit Übertrag der schwierigsten Kompetenzkategorie und Gruppe.

generelle Übersicht und Zusammenfassung von ausgewählten Leistungsdaten für alle Benutzer, die einer Lehrkraft zugeordnet sind – unterteilt in Klassen und Schüler. Eine Schülerin beziehungsweise ein Schüler stellt dabei eine einzelne Zeile der Übersichtstabelle dar. Das äußerst rechte Element dieser und nachfolgender Aufstellungen, zugehörig zum Reiter *Details*, enthält Links (*>HIER<*) zu untergeordneten individuellen Detailübersichten separat für jede Benutzerin und jeden Benutzer. Nach der Wahl einer bestimmten Schülerin oder eines Schülers wird bei Auswahl einer derartigen spezifischen Detailtabelle über diesen Link die Zeile des gewählten Users der besseren Übersichtlichkeit halber mit grauem Hintergrund belegt. Dadurch sind die gewählten Zeilen der Tabellenstruktur auf einen Blick erkennbar. Zur Veranschaulichung siehe Abbildung 4.5, in welcher die Statistikauswertung für den Benutzer Max Musterschüler mit zwei Detailtabellen zu sehen ist. Da in der ersten Tabelle die Problemkategorie *OHNE Übertrag* ausgewählt wurde (diese Zeile ist mit grauem Hintergrund versehen), ist darunter genau diese Aufstellung (siehe Überschrift *OHNE Übertrag*) mit den dazu passenden Unterteilungen inklusive Benutzerdaten aufgelistet. Weitere Untertabellen führen bis zur atomaren Auflistung der einzelnen Usereingaben. Eine ausführliche Behandlung aller aufgestellten Daten der Statistikauswertung findet in Abschnitt 4.7 statt.

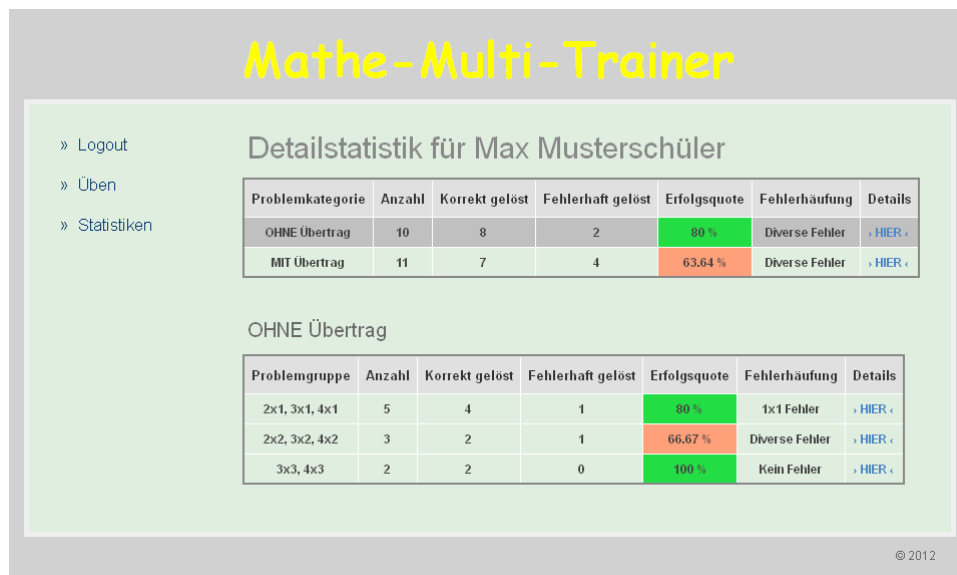


Abbildung 4.5: Beispiel für die Statistikauswertung der Lernapplikation nach Selektion eines bestimmten Users.

## 4.4 Generierung der Beispiele

Die Ermittlung von mehrstelligen Multiplikationen beruht im Wesentlichen auf einer geeigneten Berechnung von Multiplikand und Multiplikator, welche die Grundlage einer Multiplikationsrechnung ausmachen. *Geeignet* bedeutet in diesem Fall etwa die Vermeidung von doppelten Ziffern in einem Faktor oder die Umgehung der Ziffer 0 in Faktoren mit einer bestimmten Stellenanzahl, als Beispiele. Je nach Kompetenzkategorie und Problemgruppe ergeben sich so unterschiedliche Anforderungen für

Problemstellungen. Aufgabe des Algorithmus zur Generierung der Multiplikationen ist es, genau diese verschiedenen Bedingungen für eine kompakte Ermittlung von mehrstelligen Multiplikationen zu berücksichtigen, um in weiterer Folge je nach gewählter Problemgruppe die bestmöglichen Beispiele zu erstellen. Im nächsten Unterabschnitt 4.4.1 werden Ideen, Grundlagen und Entwicklungsschritte behandelt, welche alle einen wichtigen Beitrag bei der Erstellung des Algorithmus geleistet haben. Darauf aufbauend erfolgt die Vorstellung der daraus gewonnenen und im Programm eingesetzten Rechenvorschrift sowie die Präsentation von dessen Funktionsweise.

#### 4.4.1 Entwicklung des Algorithmus

Ein integrales Charakteristikum des Algorithmus zur Generierung von mehrstelligen Multiplikationen soll eine universelle Beispielerstellung sein. Dies wird durch die Miteinbeziehung von Zufallszahlen in die Rechnungsermittlung erreicht. Damit wird die Generierung von Aufgaben äußerst vielfältig und unvorhersagbar – eine zentrale Eigenschaft des Algorithmus. Code-Listing 4.1 zeigt die Realisierung dieses Merkmals in der Implementierung des Prototyps. In Zeile 1 wird dabei zuerst eine beliebige Zahl im Bereich von 0 bis 100 durch einen Zufallszahlengenerator, dargestellt durch die PHP-Funktion `mt_rand()`, erstellt. Anhand der Division durch 100 erhält man eine Zufallswahrscheinlichkeit, die sich zwischen 0 und 100 Prozent bewegen kann. Gemäß der Wahrscheinlichkeitstheorie handelt es sich hierbei um eine Zahl zwischen 0 und 1, wie im Code ersichtlich. Je nach berechnetem zufälligen Prozentwert können Faktoren nur bestimmte Ziffern belegen. Eine Ziffernmenge wird dabei durch einen festgelegten Prozentsatz definiert. In den Zeilen 3 bis 8 der behandelten Code-Aufstellung ist dieser Vorgang als Exempel gelistet: der Variable `$multiplier` wird zu jeweils 40% die Ziffer 1 oder 3 zugewiesen, zu 20% die Ziffer 2. Dieser Prozentwert ist nicht mit der Zufallswahrscheinlichkeit zu verwechseln. Erhält man als Zufallswahrscheinlichkeit also einen Wert von 55% (`$random_p = 0.55`), dann enthält der Multiplikator den Wert 2 – dies ist zu 20% möglich. Die Abfragen der Zufallswahrscheinlichkeit inklusive Zuweisungen im Code-Beispiel können beliebig angeordnet sein, solange die vordefinierten Prozentwerte mit den ihr zugewiesenen Zahlenwerten übereinstimmen. Die Maxime der Zufälligkeit bleibt immer erhalten.

---

```
1 $random_p = mt_rand(0,100)/100;
2
3 if ($random_p <= 0.40)
4     $multiplier = 1;
5     elseif (($random_p > 0.40) && ($random_p <= 0.60))
6         $multiplier = 2;
7     elseif (($random_p > 0.60) && ($random_p <= 1))
8         $multiplier = 3;
```

---

Listing 4.1: Beispiel zur Berechnung von Zufallszahlen des Generier-Algorithmus.

Prinzipiell wird die Ermittlung von übertragslosen Multiplikationen und jenen mit Übertrag getrennt behandelt, da sowohl für Multiplikand wie Multiplikator andere Definitionsmengen und Einschränkungen vorliegen. Beiden gemeinsam ist die generelle Funktionalität des Algorithmus, nämlich das Werfen von Zufallszahlen und die darauf aufbauende Ermittlung der Faktoren.

### Multiplikationen ohne Übertrag

Die erste und einfachste Problemgruppe bilden Multiplikationen mit zweistelligen Multiplikatoren und einstelligen Multiplikatoren. Daher wurde zur Erstellung des Algorithmus damit begonnen, für diese erste Problemgruppe einen passenden Multiplikator zu generieren. Ausgehend von diesem Wert erfolgt anschließend die Berechnung des entsprechenden Multiplikanden. Der Definitionsbereich von einstelligen Multiplikatoren für die angesprochene Aufgabengruppe umfasst alle Ziffern von 0 bis 9. Es muss jedoch darauf geachtet werden, dass höhere Ziffernwerte den noch zu ermittelnden Multiplikatoren einschränken. Er darf nur Werte enthalten, die zu keiner Übertragsrechnung führen. Ausgenommen ist hier und in den weiteren Erwähnungen des Multiplikanden immer die äußerst linke Stelle des Faktors. Diese kann beliebig gewählt werden, da kein weiteres Teilprodukt folgt, dem ein Übertrag aufsummiert werden könnte. Multiplikatorwerte von 5 bis 9 haben demzufolge Multiplikatoren mit der Beschränkung auf die Werte 0 und 1 als Ergebnis. Würde der Multiplikand beispielsweise die Ziffer 2 enthalten, so würde die Multiplikation mit einem Multiplikator von 5 zum Produkt 10 führen, das aber einen Übertrag von 1 enthält. Die Beschränkung der Werte muss also sichergestellt werden. Außerdem dürfen derartige höhere Ziffernwerte für Multiplikatoren nur geringfügig auftreten, da sich anderenfalls immer dieselben wenigen Multiplikatorenwerte wiederholen. Dies ist auf keinen Fall zielführend, noch dazu sollten in Faktoren die Ziffern 0 und 1 ebenso wenig oft vorkommen, um zu einfache Multiplikationen zu vermeiden. All diese Überlegungen wurden zusammengeführt. Nach weitreichenden Tests konnte für die erste Problemkategorie, bestehend aus einstelligen Multiplikatoren und zweistelligen Multiplikatoren, folgende in Tabelle 4.2 dargestellte prozentuelle Verteilung für die möglichen Werte des Multiplikators gefunden werden – eine universelle Beispielgebung vorausgesetzt:

Ziffernwert	Auftrittswahrscheinlichkeit
1	5%
2	32%
3	32%
4	21%
5 - 9	10%

Tabelle 4.2: Ziffernwerte mit entsprechenden Auftrittswahrscheinlichkeiten für einen Multiplikator der Problemgruppe 1.

Die Tabelle zeigt, dass die Wahrscheinlichkeit am größten (64%) ist, eine der beiden Ziffern 2 und 3 als Multiplikator zu erhalten. Dies hat damit zu tun, dass diese beiden Zahlen von allen möglichen den vielseitigsten Multiplikatoren und damit eine überaus vielfältige Gesamtmultiplikationsrechnung zulassen, was gemeinhin ein bedeutsames Ziel des Algorithmus darstellt. Auffallend ist die hohe Auftrittswahrscheinlichkeit von 10% für die Zahlen 5 bis 9. Dies ist ein Sonderfall der ersten Problemgruppe, da für zweistellige Multiplikatoren praktisch nur deren Einerstelle für eine Übertragsrechnung maßgeblich ist. Daher kann diese einzelne Stelle bei auftretenden hohen Ziffernwerten zu einem gewissen Grad vernachlässigt werden. Dieselbe Sichtweise erklärt die Auftrittswahrscheinlichkeit

von 21% für Ziffer 4. Für Multiplikatoren mit mehr als zwei Stellen kann diese Vorgehensweise nicht mehr angewendet werden, wie weiter unten erläutert wird.

Der Wert des Multiplikanden baut nun auf dem soeben ermittelten Multiplikator auf. Wie zuvor erwähnt, wird dessen äußerst linke Stelle getrennt von den anderen Ziffern behandelt. Dies ist im Fall von zweistelligen Multiplikatoren der Problemgruppe 1 die Zehnerstelle. Sie wird erst nach der Ermittlung der Einerstelle festgelegt und enthält einen zufälligen Wert im Bereich von 2 bis 9. Außerdem wird sichergestellt, dass sich die Zehner- von der Einerstelle unterscheidet, es keine doppelten Werte gibt. Die Berechnung der Einerstelle zuvor ist mit mehr Aufwand verbunden. Dessen Ziffernwert hängt einzig und allein vom davor erstellten Multiplikator ab und darf bei Multiplikation keinen Übertrag ergeben. Eine erste grobe Einschätzung führte zu nachfolgendem Versuch, bei dem der gegebene Multiplikator die Einerstelle des Multiplikanden nach Tabelle 4.3 vorgibt. ZZ kennzeichnet hier eine Zufallszahl. Die Tabellenaufstellung erklärt sich anhand des nachstehenden Beispiels wie folgt: Enthält der Multiplikator den Wert 1, so ergibt sich die Einerstelle des Multiplikanden zufällig aus einer Zahl im Bereich von 0 bis 9. Besteht der Multiplikator zum Beispiel aus der Zahl 2, dann darf der entsprechende Multiplikand nur eine Ziffer von 0 bis 4 umfassen, um kein Beispiel mit Übertrag zu erzeugen.

Multiplikator	Multiplikand (Einerstelle)
1	ZZ aus [0, 9]
2	ZZ aus [0, 4]
3	ZZ aus [0, 3]
4	ZZ aus [0, 2]
5 - 9	ZZ aus [0, 1]

Tabelle 4.3: Erster Testversuch für die Ermittlung eines Multiplikanden der Problemgruppe 1.

Die Ergebnisse von 100 Testdurchgängen mit dieser Konfiguration sind in Tabelle 4.4 als Auftrittswahrscheinlichkeiten ( $p$ ) für entsprechende Multiplikatorziffern in der Spalte *Ineffiziente Auftritts- $p$*  dargestellt. Die Resultate sind äußerst unbefriedigend, im Speziellen die große Häufung der Zahlen 0 und 1 für die Einerstelle des Multiplikanden. Da dies im Widerspruch zur Zielsetzung des Algorithmus steht, wurden weitere Anpassungen vorgenommen und Testläufe durchgeführt.

Schließlich wurde eine für alle Belange zufriedenstellende Einstellung gefunden, welche in Tabelle 4.5 dargestellt ist und der Problemgruppe 1 für Aufgabenstellungen ohne Übertrag als Berechnungsvorschrift der Multiplikatoren dient. Es wurde versucht, für bestimmte Fälle, primär durch die Verminderung der Zahlen 0 und 1 im Multiplikanden, die Einfachheit der Beispiele einzuschränken und im selben Zug die Vielfältigkeit dieses Faktors durch Anheben der Auftrittswahrscheinlichkeiten für die Ziffern 2 und 3 zu steigern. Bei 100 Versuchsdurchgängen stellte sich eine akzeptable Verteilung der Ziffern für die Einerstelle des Multiplikanden ein. Diese ist in der Spalte *Finale Auftritts- $p$*  in Tabelle 4.4 wiedergegeben.

Die nächste Aufgabengruppe 2 besteht ebenfalls aus Multiplikatoren mit einer Stelle, aber aus dreistelligen Multiplikatoren. Dadurch, dass nun zwei Stellen des Multiplikanden ausschlaggebend für eine Übertragsrechnung sind, ändert sich der Algorithmus sowohl für die Ermittlung des Multiplikators



Multiplikand (Einerstelle)	Ineffiziente Auftritts- $p$	Finale Auftritts- $p$
0	25%	6%
1	24%	12%
2	20%	35%
3	17%	29%
4	10%	14%
5 - 9	4%	4%

Tabelle 4.4: Ergebnisse von 100 Testläufen zur Ermittlung von Auftrittswahrscheinlichkeiten ( $p$ ) für die Einerstelle eines Multiplikanden der Problemgruppe 1.

Multiplikator	Multiplikand (Einerstelle)
1	ZZ aus [3, 9]
2	ZZ aus [3, 4]
3	ZZ aus [1, 3]
4	2
5 - 9	ZZ aus [0, 1]

Tabelle 4.5: Finale Einstellungen für die Ermittlung eines Multiplikanden der Problemgruppe 1.

als auch für jene des Multiplikanden. Die Berechnungsvorschrift für den Multiplikator wird in Tabelle 4.6 präsentiert. Die Nichtbeachtung der Zahl 1 ergibt Sinn, da für höherstellige Multiplikanden eine Teilmultiplikation mit Eins für diese Kompetenzkategorie als zu einfach empfunden wurde. Im Gegensatz zur vorhergehenden ersten Problemgruppe sind die Auftrittswahrscheinlichkeiten von höherwertigen Zahlen (4 bis 9) verringert. Jene für die Zahlen 2 und 3 machen nun den größten Teil aus, die beiden Ziffern kommen mit einer Wahrscheinlichkeit von 80% bei Beispielen der Problemgruppe 2 als einstelliger Multiplikator zum Zug.

Die Berechnung der Multiplikanden für die Aufgabengruppe 2 geschieht ähnlich wie zuvor, unter Berücksichtigung des Multiplikators. Nun sind die Einer- und Zehnerstelle für die Beachtung der Übertragsfreiheit relevant. Die Hunderterstelle des Multiplikanden, also dessen äußerst linke Stelle, umfasst jetzt Zufallswerte im Bereich von 4 bis 9, wobei erneut keine doppelten Ziffern im Multiplikanden vorkommen dürfen. Die Generierung der zwei *inneren* Ziffernwerte des Faktors – das sind alle Ziffern außer der äußerst linken – geschieht wie in Aufstellung 4.7 dargelegt: Ausgehend vom Wert des Multiplikators wird eine Zufallszahl aus einer definierten Ziffernmenge bestimmt. Außerdem werden die gezogene Zahl und womöglich schon bestimmte Stellenwerte auf Ungleichheit geprüft. Bei Äquivalenz wird solange eine Zufallszahl gezogen, bis ein unterschiedlicher Wert vorliegt. Die Funktion *digits(multiplicand)* listet die schon ermittelten Stellen des Multiplikanden auf und soll hier zum besseren Verständnis dienen.

Ziffernwert	Auftrittswahrscheinlichkeit
1	0%
2	40%
3	40%
4	15%
5 - 9	5%

Tabelle 4.6: Vorschrift für die Auftrittswahrscheinlichkeiten eines Multiplikators aus Gruppe 2.

Multiplikator	Multiplikand ( <i>innere Stelle i</i> )
1	$ZZ \text{ aus } [3, 9] \wedge i \neq \text{digits}(\text{multiplicand})$
2	$ZZ \text{ aus } [2, 4] \wedge i \neq \text{digits}(\text{multiplicand})$
3	$ZZ \text{ aus } [1, 3] \wedge i \neq \text{digits}(\text{multiplicand})$
4	$ZZ \text{ aus } [0, 2] \wedge i \neq \text{digits}(\text{multiplicand})$
5 - 9	$ZZ \text{ aus } [0, 1] \wedge i \neq \text{digits}(\text{multiplicand})$

Tabelle 4.7: Zuweisungsregeln für die Generierung von Multiplikanden für Problemgruppe 2 und 3.

Die noch ausstehende Problemgruppe 3 in Kompetenzkategorie I behandelt Beispiele mit vierstelligen Multiplikanden. Der Multiplikator ist noch immer einstellig und wird, wie in Tabelle 4.8 dargestellt, berechnet. Nun sind die Ziffern 1 und 5 bis 9 für die Generierung des Multiplikators vollkommen ausgeklammert (0%). Dadurch werden einmal mehr zu einfache und nicht zielführende Beispiele, wie etwa die Multiplikation mit Null, aus der Aufgabenmenge ausgeschlossen. Der Multiplikator enthält nur mit der geringen Wahrscheinlichkeit von 6% einen anderen Wert (4) als die Ziffern 2 und 3. Die Ermittlung der entsprechenden Multiplikanden für Problemgruppe 3 erfolgt in derselben Art und Weise wie für Problemgruppe 2. Siehe dazu Tabelle 4.7.

Ziffernwert	Auftrittswahrscheinlichkeit
1	0%
2	47%
3	47%
4	6%
5 - 9	0%

Tabelle 4.8: Auftrittswahrscheinlichkeiten für Ziffern eines Multiplikators der Problemgruppe 3.

Die Sonderbehandlung von Multiplikatoren aus Kompetenzklasse I, abhängig von deren zugehörigen Gruppe, ist nun abgeschlossen. Für alle Aufgabenstellungen der Kompetenzkategorien II und III wurde eine einheitliche Ermittlung der Multiplikatoren für die jeweiligen Problemgruppen festgelegt. Dazu erfolgt die Generierung nicht mehr *per Hand*, sondern mit Hilfe einer Datenbank. Möglich wäre zwar auch die manuelle Ermittlung wie für einstellige Faktoren, doch bietet die Speicherung von vorgegebenen festen Werten für Multiplikatoren in der Datenbank eine einfachere Handhabung. Für einstellige Faktoren ist diese Vorgehensweise aufgrund der wechselhaften Wahrscheinlichkeitsverteilungen pro Aufgabengruppe nicht angebracht. Die gespeicherten Werte in der Datenbank werden nun zufällig, aber durch gegebene Wahrscheinlichkeitswerte definiert, gezogen. Die möglichen zweistelligen Multiplikatoren für alle übertragslosen Problemgruppen der Kompetenzkategorie II sind in Tabelle 4.9 abgebildet. Es sind auch Faktoren möglich, welche die Ziffer 1 enthalten. Aufgrund der Mehrstelligkeit der Multiplikatoren wird hierbei die Schwierigkeit der Beispiele nur zu einem geringen, vernachlässigbaren Grad herabgesetzt – im Gegensatz zum Fall mit einstelligen Faktoren. Es ist die Komplexität der Rechnung mit zweizeiligen kompletten Teilmultiplikationen einschließlich einer Gesamtaddition, welche unter diesem Aspekt eine Berücksichtigung der Ziffer 1 bei Multiplikatoren mit niedrig verteilten Auftretswahrscheinlichkeiten zulässt. Als Ganzes betrachtet werden die Werte 23 und 32 die meisten Multiplikatoren dieser Kategorie ausmachen. Wie erwähnt, gewährleisten diese Ziffern die bestmögliche Vielfalt der zu berechnenden Multiplikanden.

Multiplikator	Auftretswahrscheinlichkeit
12	~ 4.5%
13	~ 4.5%
14	~ 4.5%
21	~ 9.1%
23	~ 18.2%
24	~ 9.1%
31	~ 9.1%
32	~ 18.2%
34	~ 9.1%
41	~ 4.5%
42	~ 4.5%
43	~ 4.5%

Tabelle 4.9: Alle möglichen Multiplikatoren und deren Auftretswahrscheinlichkeiten für die Problemgruppen 3 bis 6 (Kompetenzkategorie II).

Die entsprechenden Multiplikanden ergeben sich fast ident zur Ermittlung derselben Faktoren aus Kompetenzklasse I, siehe dazu Tabellen 4.5 und 4.7. Der einzige Unterschied ist, dass nun mehrstellige Multiplikatoren auf enthaltene Ziffern hin analysiert werden müssen. Die Vorgehensweise wurde folgendermaßen festgelegt, dass ein Faktor zuerst auf die Existenz von höchstwertigen Ziffern (5 bis 9) geprüft wird. Die Untersuchung wird folglich nur bei Nichtauftreten der soeben kontrollierten Zahl

im Faktor mit stufenweise erniedrigten Werten fortgesetzt. Enthält der Multiplikator indes eine dieser Ziffern, erfolgt die Generierung des Multiplikanden entsprechend der passenden Tabelleneinträge.

Für dreistellige Multiplikatoren der Kompetenzkategorie III wurden aus demselben Grund und in gleicher Weise wie im Abschnitt davor für Kompetenzklasse II Faktoren in der Datenbank abgelegt. Diese sind in Aufstellung 4.10 inklusive der ihr zugeteilten Auftrittswahrscheinlichkeiten gelistet.

Multiplikator	Auftrittswahrscheinlichkeit
123	~ 16.7%
132	~ 16.7%
213	~ 16.7%
231	~ 16.7%
312	~ 16.7%
321	~ 16.7%

Tabelle 4.10: Alle möglichen Multiplikatoren einschließlich der Auftrittswahrscheinlichkeiten für die Problemgruppen 7 und 8 (Kompetenzkategorie III).

Vorhergehende Tabelle 4.10 zeigt die Einschränkung von dreistelligen Multiplikatoren bei Beispielen ohne Übertrag auf die drei Ziffern 1, 2 und 3. Dies stellt eine Herabsetzung der Vielfältigkeit von Multiplikationen für die Problemgruppen dieser Kategorie dar. Jeder einzelne Multiplikator enthält die drei genannten Ziffern, nur in anderer Anordnung. Für Faktoren dieser Art stellt diese Lösung dennoch die einzig sinnvolle dar. Die Zahl 0 als Ziffernwert des Multiplikators wurde von vornherein in der Definitionsmenge nicht zugelassen, um die Zweckmäßigkeit von Aufgaben nicht in Frage zu stellen. Also bleibt nurmehr die Ziffer 4 als alleinige Alternative übrig. Der Grund, warum auch diese Ziffer nicht toleriert wird, liegt in der Generierung der Multiplikanden. Die Ziffer 4 im Multiplikator lässt nur drei unterschiedliche Ziffernwerte im Multiplikanden zu, was bei Faktoren mit vier Stellen unweigerlich zu einer doppelt vorhandenen Ziffer führen würde. Genau das soll aber verhindert werden. Daher wurde die Beschränkung des Multiplikators auf drei Ziffern auf Kosten der Vielseitigkeit für übertragslose Beispiele in Kompetenzkategorie III beschlossen. Für dreistellige Multiplikanden würde prinzipiell auch die Ziffer 4 im Multiplikator zu keiner Verdopplung von Ziffernwerten führen. Eine gewissenhafte Abschätzung ergab jedoch auch für diese Aufgabengruppe dieselbe Einschränkung von Multiplikatoren wie für Beispiele mit vierstelligen Multiplikanden. Dadurch, dass jeder Faktor dieselben drei Ziffern enthält, nur in anderer Reihenfolge, sind alle sechs möglichen Multiplikatoren in deren Auftrittswahrscheinlichkeiten gleichverteilt.

Die Ermittlung der drei- und vierstelligen Multiplikanden für Kompetenzklasse III ist ident mit der Vorgehensweise wie im Abschnitt davor für Kategorie II erläutert. Als Anschauung dazu dient Tabelle 4.7.

### Multiplikationen mit Übertrag

Die Beispielermittlung von Multiplikationen mit Übertragsrechnung wird von der Lernapplikation als diejenige Multiplikation definiert, die für jedes einzelne enthaltene Teilprodukt einen Übertrag garantiert. Eine Mischung von übertragslosen und übertragsbehafteten Teilrechnungen einer Aufgabenstellung ist nicht möglich. Somit wird vorgegeben, dass bei Multiplikationen mit Übertrag theoretisch jede Übertragszelle des Antwortformulars eine Befüllung verlangt.

Die Generierung von Faktoren für Aufgabenstellungen mit Übertrag ist um einiges einfacher als bei der übertragslosen Beispielermittlung und bietet einen großen Spielraum in der Verwendung von variablen Ziffernwerten. Darüber hinaus wurde ein Algorithmus entworfen, der alle acht Problemgruppen abdeckt. Eine Sonderbehandlung für Faktoren aus bestimmten Gruppen ist nicht mehr nötig, da die Vorgabe der Übertragsrechnung keine großflächigen individuellen Einschränkungen in den Aufgabengruppen verlangt. Multiplikatoren für diese Gruppen berechnen sich abermals durch definierte Prozentwerte, welche das Auftreten von verwendbaren Ziffern regeln und bestimmen. Die Aufstellung der Ziffern einschließlich deren Auftrittswahrscheinlichkeiten für übertragsbehaftete Multiplikatoren ist in anschließender Tabelle 4.11 gegeben. Dessen Prozentwerte resultieren aus eingehenden Testläufen, ähnlich wie zuvor für Beispiele ohne Übertrag. Es ist ersichtlich, dass Ziffern mit niedrigem Wert (1 bis 4) für die Ermittlung des Faktors weniger maßgebend sind als jene mit höherem Wert (5 bis 9). Die Zahl 1 ist aus der Definitionsmenge der möglichen Ziffern für Multiplikatoren jedenfalls ausgeschlossen (0%), da dieser keine einstellige Ziffer entspricht, um durch gemeinsame Multiplikation einen Übertrag zu gewährleisten. Der Wertebereich von 5 bis 9 umfasst die Gruppe der Zahlen mit der höchsten Wahrscheinlichkeit des Auftretens. Dabei handelt es sich um jene Elemente, die eine Übertragsrechnung auf vielfältige Weise möglich machen, indem der jeweilige Multiplikand alle einstelligen Ziffern außer die für die Beispielgebung generell unwesentlichen Zahlen 0 und 1 beinhalten darf, um eine Multiplikation mit Übertrag zu erhalten. Wird diese Gruppe durch Werfen einer Zufallszahl getroffen, so ermittelt das Programm wiederum per Zufall eine Zahl aus genau diesem Bereich. Für mehrstellige Faktoren wiederholt sich diese Vorgehensweise entsprechend, wobei bei der Generierung darauf geachtet wird, dass jede Ziffer des Multiplikators nur einmal auftritt.

Ziffernwert	Auftrittswahrscheinlichkeit
1	0%
2	7%
3	10%
4	15%
5 - 9	68%

Tabelle 4.11: Die möglichen Ziffern inklusive Auftrittswahrscheinlichkeiten von Multiplikatoren für übertragsbehaftete Beispiele.

Die Berechnung der Multiplikanden für Multiplikationen mit Übertrag obliegt ebenso einem einheitlichen Verfahren für alle Problemgruppen und ähnelt dem Ablauf der übertragslosen Beispielermittlung. Der Unterschied liegt in den definierten Ziffernbereichen für Multiplikanden bei bestimmten Mul-

Multiplikatorelementen und dem Überprüfungsprozess der einzelnen Werte. Für die Generierung der Multiplikatoren sind nun die niedrigsten Ziffernwerte eines Multiplikators, beginnend mit der Ziffer 1, entscheidend – im Gegensatz zu den höchstwertigen Elementen im Fall der übertragslosen Aufgabenstellungen. Die Multiplikatoren werden anfangs dahingehend kontrolliert. Wird die Ziffer im Multiplikator nicht gefunden, erfolgt eine stetige Erhöhung des Wertes, bis die überprüfte Ziffer im Faktor gefunden wird. Der erste erfolgreiche Vergleich eines Elementes hat zur Folge, dass nun gemäß Tabelle 4.12 ein passender Multiplikand generiert werden kann. Die Übersicht bildet die Ermittlung der *inneren* Stellen des Multiplikanden durch Ziehung einer Zufallszahl (ZZ) aus einer Definitionsmenge für gegebene Multiplikatorelemente ab. Die erste Zeile der Tabelle ist in hellgrauer Schrift verfasst, da Multiplikatoren grundsätzlich keinen Ziffernwert von 1 enthalten dürfen, wie im Abschnitt davor festgelegt wurde (vergleiche Tabelle 4.11). Ferner ist es nun möglich, dass der generierte Multiplikand gleiche Ziffernwerte enthält, da durch den Vorgang der Übertragsaddition auch bei identischen Teilmultiplikationen andere Denkprozesse gefordert werden und somit der implizierten Vereinfachung entgegengesteuert wird. Die äußerst linke Stelle des Multiplikanden wird getrennt von den übrigen Stellenelementen ermittelt und kann eine Zahl im Bereich von 2 bis 9 annehmen, wobei diese garantiert ungleich der bereits generierten *inneren* Stellen des Faktors gewählt wird.

Multiplikator (Ziffer)	Multiplikand ( <i>innere</i> Stelle)
1	ZZ aus [5, 9]
2	ZZ aus [5, 9]
3	ZZ aus [4, 9]
4	ZZ aus [3, 9]
5 - 9	ZZ aus [2, 9]

Tabelle 4.12: Generiervorschrift für Multiplikatoren bei Aufgabenstellungen mit Übertrag.

## 4.5 Evaluierung der Eingaben

Die Auswertung der Benutzereingaben für eine Multiplikationsrechnung erfolgt sofort nach Betätigung des Antwortbuttons durch eine Anwenderin oder einen Anwender, vergleiche dazu Abbildung 4.2. Für gewöhnlich hat der User zuvor die einzelnen Antwortfelder der aktuell im Browser ausgegebenen Aufgabenstellung gemäß den Teilergebnissen der Multiplikation befüllt. Der Inhalt aller vorhandenen Inputzellen wird infolgedessen an das Programm geschickt. Dessen Auswertungsalgorithmen analysieren die gesammelten Daten, ziehen Schlussfolgerungen und veranlassen Sicherungen von brauchbaren und neu generierten Daten.

Der Evaluierungsprozess steuert einerseits die wiederholte Anzeige der Multiplikationsrechnung einschließlich aller Benutzereingaben durch Angabe der Korrektheit der Lösungsfindung beziehungsweise durch Fehlermarkierung von falsch eingegebenen Zellen (siehe Darstellungen 4.3 und 4.4). Der wichtigste Teil des Auswertungsvorgangs findet im Hintergrund statt. Dieser behandelt das automatische Auffinden und Kennzeichnen von typischen Fehlern anhand der Antworten der Benutzer.

Die berechneten Werte werden in Datenbanktabellen gespeichert und dienen der Statistikauswertung für Lehrkräfte, welche in Abschnitt 4.7 behandelt wird, als Datenquelle. Die Art der Multiplikationsrechnung bestimmt im Programm den Algorithmus zur Datenevaluierung. Grundsätzlich gibt es zwei wichtige Berechnungsvorschriften, wobei sich der eine für übertragslose Beispiele und der andere für Beispiele mit Übertrag verantwortlich zeichnet. Die Fehlerbestimmung gestaltet sich für Aufgabenstellungen ohne Übertragsrechnung zu einem hohen Grad einfacher als für Letztgenannte. Dies hat hauptsächlich damit zu tun, dass für übertragslose Problemstellungen nur eine einzige Eingabezelle pro Teilprodukt untersucht werden muss, wohingegen für Übertragsbeispiele zwei Felder pro Teilergebnis der Multiplikation zur Berechnung herangezogen werden. Dies macht einen beträchtlichen Zuwachs an Berechnungsschritten aus. Von der Behandlung von Überträgen ausgenommen sind immer die beiden äußerst linken Eingabezellen einer Ergebniszeile, da diese verständlicherweise nie eine Übertragszelle aufweisen – der Übertrag macht in diesem Fall die Zehnerstelle des Teilproduktes aus und muss dementsprechend in das passende Zehnerfeld eingetragen werden. Es wird ohnedies kein Übertragsfeld angezeigt.

Wichtigstes Ziel des Evaluierungsprozesses ist die einwandfreie Auffindung von Fehlern aus den Benutzereingaben. Durch eine Analyse und intensive Testung von diversen Multiplikationsaufgaben wurden typische Fehler ausgemacht, welche ohne Zweifel richtig interpretiert und rückverfolgt werden konnten. Im nächsten Schritt wurden diese Fehler benannt und für die Lernapplikation im Code definiert. Zusätzlich wurden unterschiedliche sogenannte OK-Zustände für Problemstellungen gefunden und in gleicher Weise festgelegt. Ein OK-Zustand ergibt sich aus der richtigen Lösung einer Multiplikation oder Teilmultiplikation. In vielen Fällen werden zu dieser schlichten Information, wenn möglich, Zusatzhinweise geliefert. Diese können zum Beispiel die Information beinhalten, dass ein untersuchtes Teilprodukt korrekt gelöst wurde, obwohl Folgefehler in die Berechnung miteinbezogen wurden. Ein Auswertungsalgorithmus erkennt anhand der Antworteingaben der Benutzer, ob es sich bei der Beantwortung einer Multiplikation oder Teilmultiplikation um eine erfolgreiche oder inkorrekte Lösung handelt. Ferner erkennt die Berechnungsvorschrift bei unrichtiger Lösung, welcher Fehler aus der definierten Fehlermenge vorliegt. Bei richtigem Ergebnis liefert die Berechnung einen passenden OK-Status aus der definierten OK-Zustandsmenge. Der entwickelte Algorithmus enthält Berechnungsvorschriften zur Evaluierung von übertragslosen und übertragsbehafteten Problemstellungen, er erkennt darüber hinaus typische Additionsfehler, welche bei Gesamtadditionen in mehrzeiligen Ergebnisfindungen auftreten können – dieser Fall ist bei mehrstelligen Multiplikatoren gegeben. Der komplexe Algorithmus zeichnet sich durch die exakte Bestimmung von definierten Fehlern bei Multiplikationen aus beliebigen Problemgruppen aus. Diese Mächtigkeit der implementierten Berechnungsvorschrift ist für Lehrkräfte immens vorteilhaft, da die ermittelten Evaluierungsdaten in der Statistikauswertung uneingeschränkt zur Verfügung stehen und die Leistungsfähigkeit von Schülern auf einen Blick und ohne eigene Untersuchungen greifbar ist.

Die definierten OK-Zustände des Mathe-Multi-Trainers sind in Tabelle 4.13 festgehalten, die gefundene Fehlermenge ist in Tabelle 4.14 gelistet. Auf die spezifischen Fehler- und OK-Zustände wird gesondert in den nächsten zwei Abschnitten eingegangen. Alle in den Tabellen dargestellten Zustandskonstellationen ergeben sich aus der Lösung einer einzigen Teilmultiplikation, bestehend aus ausschließlich einstelligen Faktoren, deren Summe in der Regel eine gesamte Aufgabenstellung ausmacht. Als Vermerk sei erwähnt, dass Einzelberechnungen von Problemstellungen mit Übertrag auch Lösungen der jeweils letzten Teilmultiplikation – das sind deren Überträge – in die aktuelle Fehlerberechnung miteinbeziehen. Hinzu kommen für bestimmte Rechnungsarten Fehler- oder OK-Zustände

für Additionen von zeilenweisen Teilergebnissen. Zum Beispiel umfasst eine Multiplikation der Problemgruppe 4 mit jeweils zweistelligen Faktoren vier separate Teilmultiplikationen, aufgeteilt in zwei Zeilen, und eine Addition der beiden Ergebniszeilen. Der Auswertungsalgorithmus wird auf alle fünf Berechnungsschritte angewendet und konstruiert abhängig von den Lösungseingaben Zustände entweder aus Aufstellung 4.13 oder aus Tabelle 4.14.

Das Vorkommen von Zuständen als Übersichts-Status beziehungsweise als Übersichts-Fehler (vergleiche hierzu den ersten Reiter der behandelten Tabellen) ist nur bei allumfassenden Rechnungsinterpretationen möglich. Diese können als eine Zusammenfassung von aufgetretenen Fehlern oder OK-Zuständen entweder für eine Reihe von Multiplikationen oder für einzelne Aufgabenstellungen betrachtet werden. Bei Letzteren werden die individuellen Teilzustände der einstelligen Teilprodukte zur Schlussfolgerung herangezogen. Treten mehr als zwei verschiedene Fehlerarten bei derartigen Untersuchungen auf, so ergibt sich der Fehlerstatus zu `ERR_MISC`. Bei fehlerfreien Lösungen berechnet sich der Status zu `OK` oder auch `NO_ERROR`. Wird ein Antwortfeld irrtümlich mit Symbolen oder Buchstaben befüllt, so erscheint im Browser die Fehlermeldung *Nur Zahlen eingeben!* in roter Schriftfarbe. Eine Auswertung passiert nicht, dieselbe Aufgabenstellung bleibt zum erneuten Lösungsversuch angezeigt.

Findet eine korrekte Lösungseingabe für Beispiele aller Art statt, gibt der Algorithmus als Feedback den Status `OK` zurück. Zur Veranschaulichung dient Abbildung 4.3, welche eine einwandfreie Lösung für eine Multiplikation der Problemgruppe 6 zeigt. Insgesamt haben neun individuelle Berechnungsschritte den Zustand `OK` als Resultat, acht davon für die richtigen Teilprodukte und einer für die fehlerfreie Gesamtaddition. Als Zusammenfassung der Aufgabenstellung ergibt sich der Übersichts-Status gleichermaßen zu `OK` oder `NO_ERROR`, je nach gewählter Überblicksart.

#### 4.5.1 Auswertung von Multiplikationen ohne Übertrag

Für Multiplikationen sind je nach Rechnungsmodus unterschiedliche Fehler und OK-Zustände definiert. Die verschiedenen Modi sind in den Statustabellen im Reiter *Vorkommen* differenziert aufgestellt. Der positive Zustand `OK` kann naturgemäß sowohl bei Multiplikationen mit und ohne Übertrag, als auch bei Gesamtadditionen in Multiplikationsrechnungen vorkommen. In gleicher Weise kann ein Eingabefehler (`ERR_INPUT`) in allen drei grundlegenden Rechnungsarten auftreten. Dieser Fehler wird anhand einer Aufgabenstellung der Problemgruppe 5 in Darstellung 4.6 gezeigt. Die fehlenden Eingaben der ersten Ergebniszeile und der Endsumme markieren hier den angesprochenen Fehler. Dieser wird vom Algorithmus als solcher berechnet und in der Datenbank abgespeichert. Für Benutzer findet im Ergebnismodus (siehe Abbildung) keine spezifische Fehlerausgabe statt, stattdessen werden nur die falsch befüllten Antwortzellen durch eine rote Umrandung als Unrichtigkeit kenntlich gemacht – ein Mehr an Informationen ist für Schüler unangemessen. Bleibt das jeweilige äußerst linke Feld einer Ergebniszeile unbefüllt, dann führt dies nicht zu einem Eingabefehler, da es sich bei diesem Element um die Zehnerstelle eines Teilproduktes handelt und dessen Ergebnis eben auch einstellig sein könnte. Eine Summenzeile verhält sich mit ihren beiden höchstwertigen Stellen ähnlich. Genau diese Situation ist in behandelter Beispielabbildung 4.6 in der ersten und letzten Ergebniszeile gegeben. Wie zu sehen, werden für die unbefüllten Felder links außen keine Eingabefehler berechnet. Diese werden daher richtigerweise als korrekt markiert.



Vorkommen	OK-Status	Kurzbeschreibung
Beispiele ohne Übertrag	OK	OK ( <i>fehlerfrei</i> )
Beispiele mit Übertrag	OK	OK ( <i>fehlerfrei</i> )
	OK_MENTAL	OK (Kopfrechnung)
	OK_NOCARRY	OK (Kopfrechnung mit Nachfolgefehler)
	OK_ERR	OK (inkludiert Folgefehler)
Addition	OK	OK ( <i>fehlerfrei</i> )
	OK_END_ERR	Addition OK (inkludiert Folgefehler)
Übersichts-Status	OK	OK ( <i>fehlerfrei</i> )
	NO_ERROR	Kein Fehler

Tabelle 4.13: Alle feststellbaren OK-Zustände des Mathe-Multi-Trainers und deren Vorkommen.

Vorkommen	Fehler-Status	Kurzbeschreibung
Beispiele ohne Übertrag	ERR_INPUT	Eingabefehler
	ERR_1x1	1x1 Fehler
	ERR_REV_IN	Eintragungsfehler (Vertauschung)
Beispiele mit Übertrag	ERR_INPUT	Eingabefehler
	ERR_1x1	1x1 Fehler
	ERR_REV_IN	Eintragungsfehler (Vertauschung)
	ERR_NOADD	Übertrag-Addierfehler (Korrektes 1x1 – kein Addieren)
	ERR_ADD	Übertrag-Addierfehler (trotz korrekter Antwort)
	ERR_UNKNOWN	Unbekannter Fehler
Addition	ERR_INPUT	Eingabefehler
	ERR_END_ADD	Additionsfehler
	ERR_END_OK	Additionsfehler (trotz korrekter Antwort)
Übersichts-Fehler	ERR_MISC	Diverse Fehler

Tabelle 4.14: Alle feststellbaren Fehler des Mathe-Multi-Trainers und deren Vorkommen.



Abbildung 4.6: Zwei Eingabefehler (ERR\_INPUT) beim Lösen einer Multiplikationsaufgabe.

Das Rechnungsbeispiel in Darstellung 4.7 zeigt in den ersten zwei Ergebniszeilen einwandfrei feststellbare Einmaleins-Fehler (ERR\_1x1) für Aufgabenstellungen ohne Übertrag. Das erste falsch befüllte Antwortfeld hält ein einstelliges Teilprodukt (9), während die zweite fehlerhafte Eingabe auf einem ein- oder zweistelligen Teilprodukt beruht. Hier zeigt sich wieder die Sonderbehandlung der zwei äußersten linken Eingabefelder. Im Beispiel wurde das Teilprodukt irrtümlicherweise mit 19 angegeben, obwohl 9 die richtige Antwort gewesen wäre. Darüber hinaus repräsentiert die Summenzeile, gleichzusetzen mit dem Gesamtergebnis der Multiplikation, zwei falsch markierte Eingabezellen. Dabei handelt es sich jedoch nicht um eine fehlerhafte Addition der beiden ersten Ergebniszeilen. Die beiden falsch gesetzten Felder ergeben sich aus den Einmaleins-Fehlern der Teilmultiplikationen von zuvor. Somit beinhaltet die an sich korrekte Addition Folgefehler von früheren Berechnungen, womit die beiden Fehler im Gesamtergebnis erklärt sind. Der Auswertungsalgorithmus kennzeichnet einen derartigen Fehler mit dem Additions-Status OK\_END\_ERR.

Die letzte der drei Fehlergattungen für übertragslose Multiplikationen behandelt die vertauschte Eintragung eines zweistelligen Teilproduktes und wird in Abbildung 4.8 wiedergegeben. Dieser Fehler (ERR\_REV\_IN) kann nur in den beiden höchststelligen Elementen einer Ergebniszeile auftreten. Die vertauschte Eintragung bedeutet, dass prinzipiell richtig gerechnet wurde, das Ergebnis der Lösungsfindung jedoch falsch in die Antwortzellen eingetragen wurde. Der Fall, dass Schüler einen Einmaleins-Fehler begehen und dessen falsches Ergebnisprodukt genau der vertauschten Eintragung der korrekten Lösung entspricht, wird als sehr unwahrscheinlich eingestuft und mündet somit immer in einem ERR\_REV\_IN Fehler. Das Beispiel zeigt außerdem eine richtige Gesamtlösung der Multiplikationsaufgabe, dargestellt durch den Text *Richtiges Endergebnis!*. Dieses Resultat kann allerdings nur bei einer fehlerhaften Addition zustande kommen. Eine korrekte Addition der beiden Antwortzeilen würde Fehler im Gesamtergebnis zur Folge haben. Daher ermittelt der Algorithmus für diesen Fall einen Additions-Fehler mit dem Hinweis, dass die fehlerbehaftete Aufsummierung (zufällig) das



Abbildung 4.7: Demonstration von 1x1 Fehlern (ERR\_1x1) und einer korrekten Addition mit Folgefehlern (OK\_END\_ERR) beim Lösen einer übertragslosen Multiplikation.

richtige Endergebnis erzeugt hat. Diese Angaben sind im Fehler-Status `ERR_END_OK` gespeichert und dienen hauptsächlich zur Information. Ein derartiger Fehler kann unter Umständen darauf hinweisen, dass das Endergebnis Resultat von nicht selbst erbrachter Leistung ist (eventuelle Zuhilfenahme von Taschenrechnern, als Beispiel) und die Schülerin oder der Schüler dieses Aufgabengebiet aufgrund der Fehler davor eigentlich nicht beherrscht.

#### 4.5.2 Auswertung von Multiplikationen mit Übertrag

Multiplikationsaufgaben mit Übertrag bieten eine breiter gefächerte Auswahl und Behandlung von möglichen Fehlern und OK-Zuständen als Problemstellungen ohne Übertrag. Die Übertragungselemente spielen dabei die entscheidende Rolle. Das erste Beispiel dieser Kategorie ist in Abbildung 4.9 dargestellt und zeigt bereits die Komplexheit der Zustandsberechnung für diese Beispielkategorie. Die erste Antwortzeile der Berechnung repräsentiert eine komplett richtige Lösung von Teilmultiplikationen ohne Eintragung der Übertragungszellen. Dies bedeutet, dass die Addition der Überträge quasi im Kopf berechnet wurde. Daraus erkennt der Algorithmus den Zustand `OK_MENTAL`, auf den diese Beschreibung zutrifft. Die Übertragungsfelder werden zwar als inkorrekt ausgefüllt gekennzeichnet, dies stellt in diesem Fall aber keinen Rechenfehler dar. Die Markierung bei Nichteintragung soll Schüler zum Ausfüllen animieren, um bestmöglichen Lernerfolg sicher zu stellen. Die zweite Antwortzeile im vorliegenden Beispiel zeigt ein denkbare Szenario bei Nichtbefüllung von Übertragungszellen. Dessen Einerstelle ist noch korrekt eingetragen, die nachfolgenden beiden Stellen sind jedoch fehlerhaft befüllt. Da das Programm durch die Nichteintragung der Überträge nicht wissen kann, ob nun die Zehnerstelle des ersten Teilproduktes falsch berechnet wurde – dies würde einem Einmaleins-Fehler gleichen – oder erst bei der zweiten Teilmultiplikation ein Fehler aufgetreten ist, wird vom Algo-



Abbildung 4.8: Demonstration von Fehlern mit vertauschter Eintragung (ERR\_REV\_IN) und dem Additionsfehler ERR\_END\_OK beim Lösen einer übertragslosen Multiplikation.

rithmus der positive Status OK\_NOCARRY ermittelt. Dieser bedeutet eine korrekte Lösungsfindung durch Kopfrechnung, jedoch mit aufgetretenen Nachfolgefehlern. Entsprechender Hinweis ist wichtig, da folglich die Information ersichtlich wird, dass eine möglicherweise falsche Lösungsfindung für die behandelte Teilrechnung nicht hundertprozentig ausgeschlossen werden kann. Die nächsten beiden fehlerhaften Eingabefelder markieren einen nicht bestimmbar Fehler (ERR\_UNKNOWN). Da nicht festgestellt werden kann, ob die falschen Lösungselemente durch Einmaleins-Fehler oder durch inkorrekte Übertragsadditionen zustande gekommen sind, tritt dieser für Lehrkräfte schlechtest mögliche Fall in Erscheinung – ein nicht klassifizierbarer Fehler. Das Gesamtergebnis der letzten Zeile in Anschauung 4.9 weist zwei falsche Antwortzellen auf. Der Fehler der Aufsummierung liegt aber in der Zehnerstelle des Resultats. Anstatt der Ziffer 4 würde eine korrekte Addition die Ziffer 3 für dieses Feld ergeben. Dieser einwandfrei feststellbare Additionsfehler wird durch den Fehler-Status ERR\_END\_ADD festgelegt.

Im vorherigen Unterabschnitt wurde der Vertauschungsfehler bereits definiert. Bei übertragsbehafteten Multiplikationen kann dieser nun nicht mehr nur in den linken äußeren Stellen von zeilenweisen Ergebnissen vorkommen. Durch die präsenten Übertragsfelder ist der Fehler praktisch überall feststellbar, sofern alle Zellen ausgefüllt werden. Der Vertauschungs-Fehler ERR\_REV\_IN ist durch die Einerstelle und das zugehörige Übertragsfeld der ersten Antwortzeile in Abbildung 4.10 für diese Multiplikationskategorie dargestellt. Hier wurde die korrekte Lösung von 42 vertauscht als 24 eingetragen. Das nächste Feld ergibt demzufolge einen Fehler (dessen Übertragselement ist durch Zufall richtig mit der Zahl 5 befüllt), wobei sich dieser durch korrektes Einmaleins mit fehlerhaftem Additionsübertrag ergeben hat. Dies wird vom Programm als eigentlich korrekte Berechnung erkannt und liefert dementsprechend den Zustand OK\_ERR zurück. In der zweiten Ergebniszeile wird aufgrund dessen ersten falschen Übertragsfeldes ein korrekt feststellbarer Einmaleins-Fehler (ERR\_1x1) ermittelt. Die

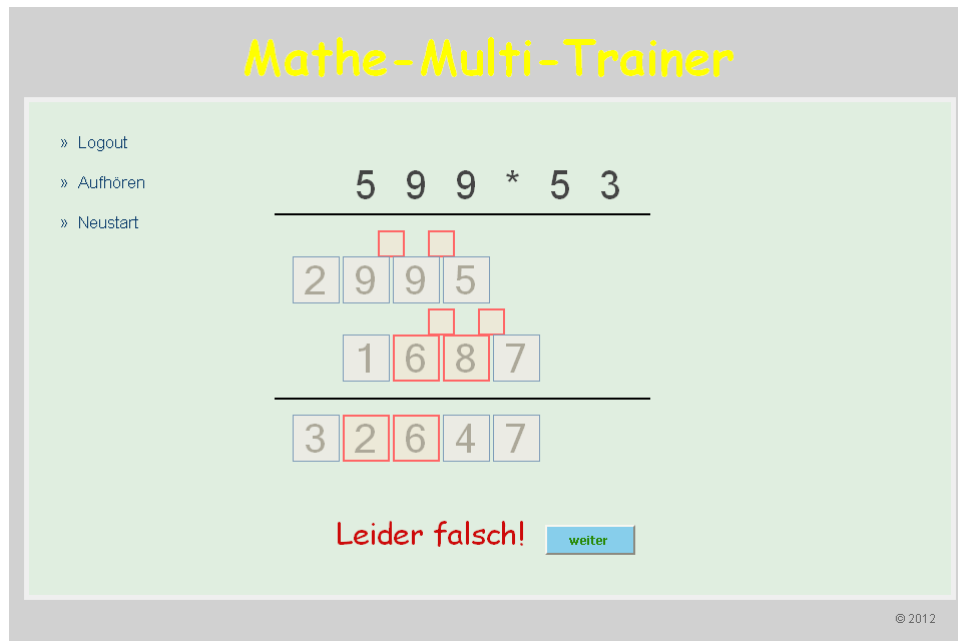


Abbildung 4.9: Demonstration der Zustände OK\_MENTAL und OK\_NOCARRY sowie der Fehler ERR\_UNKNOWN und ERR\_END\_ADD beim Lösen einer Multiplikation mit Übertrag.

Zehnerstelle dieser Zeile, ergo die folgende Eingabezeile, weist auf den ersten Blick keinen Fehler auf. Bei näherem Hinsehen ergibt sich die ‚korrekte‘ Antwort aber durch einen Addierfehler mit dem falsch eingetragenen Übertragsfeld davor. Der Fehler ERR\_ADD wird für diese Rechnungseigenschaft vom Algorithmus generiert und bezeichnet einen Übertrag-Addierfehler trotz korrekter Antwort. Als Status der Addition der beiden Ergebniszeilen wird eine fehlerfreie Berechnung erkannt, aufgrund der Fehler zuvor beinhaltet das Resultat jedoch Folgefehler (vergleiche die als falsch markierten Zellen). Dieser Eigenschaft entspricht der Additions-Zustand OK\_END\_ERR.

Abbildung 4.11 zeigt die Lösung einer Multiplikationsaufgabe aus Problemgruppe 8 ohne Aufsummierung der Überträge. Das bedeutet, dass sich in den jeweiligen Antwortzellen die korrekten Einmaleins-Berechnungen befinden, ohne jedoch nötige Additionsschritte mit Übertragsfeldern zu berücksichtigen. In der ersten Antwortzeile wurden alle Zellen befüllt, während für die folgende Zeile Kopfrechnungen ohne Eintragung und Berücksichtigung der Übertragsfelder stattgefunden haben. Beide Zeilen ergeben für jedes Feld den Fehler ERR\_NOADD, welcher die Missachtung der Übertrags-Addition kennzeichnet. Ausgenommen davon sind jeweils die Einerstellen inklusive ihrer entsprechenden Übertragsfelder. Die angesprochenen Zellen der ersten Zeile erzeugen den Status OK für die korrekte Lösung 81, diejenigen der zweiten Antwortzeile produzieren den Zustand OK\_NOCARRY, also eine korrekte im Kopf berechnete Antwort mit Nachfolgefehler. Die finale zeilenweise Addition wird, wie in diesem Fall, richtig gelöst, enthält jedoch Folgefehler durch die vorherigen falschen Berechnungen. Dies wird wieder durch den Status OK\_END\_ERR charakterisiert.

Es sei angemerkt, dass alle vorgestellten Fehler- und OK-Zustände durch Multiplikationen der Kompetenzstufe II wiedergegeben wurden. Der Grund dafür liegt in der Darstellungsform dieser

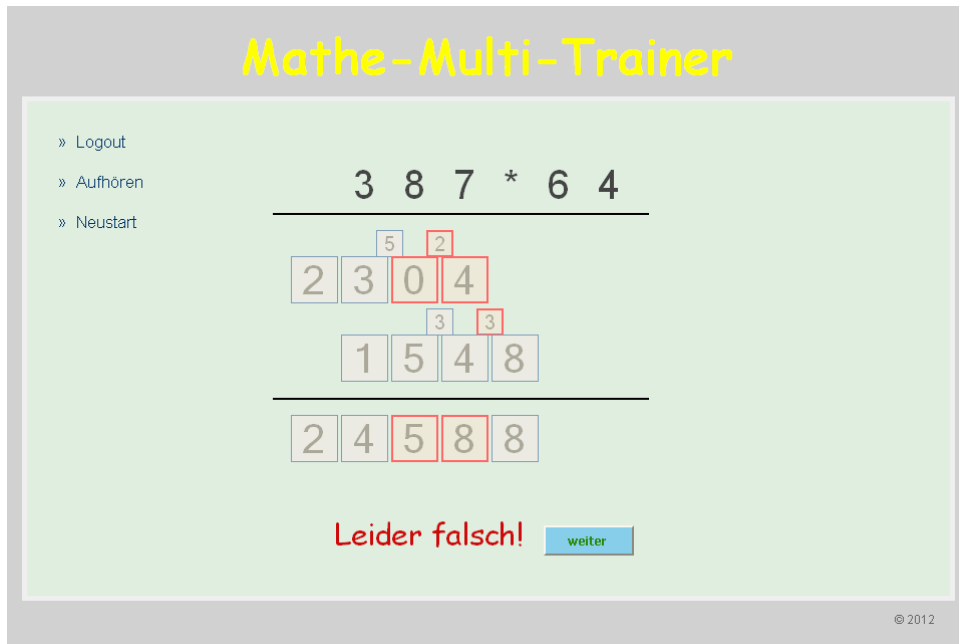


Abbildung 4.10: Demonstration der Zustände OK\_ERR und OK\_END\_ERR sowie der Fehler ERR\_REV\_IN, ERR\_1x1 und ERR\_ADD beim Lösen einer Multiplikation mit Übertrag.

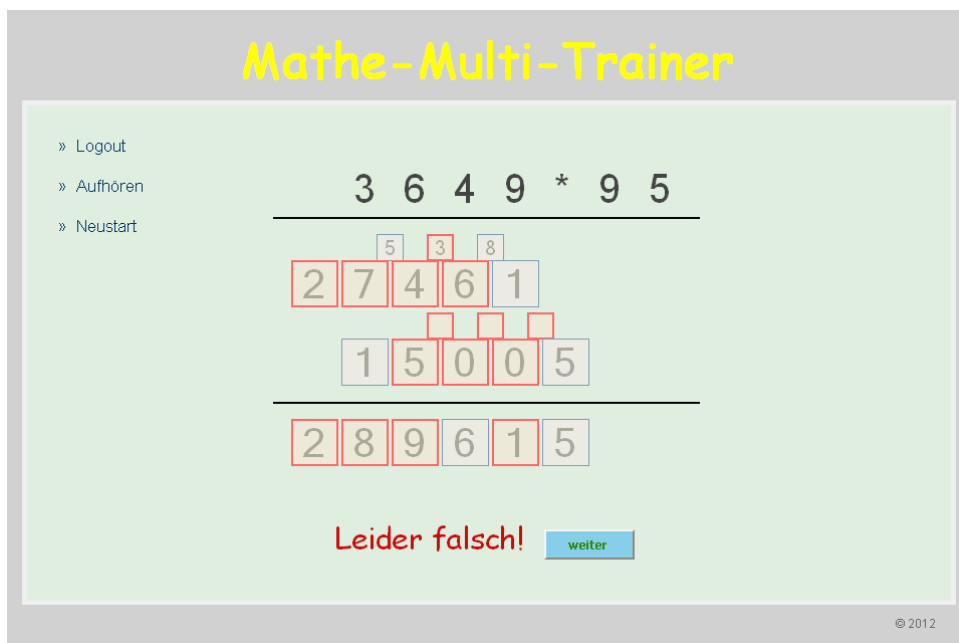


Abbildung 4.11: Demonstration der Zustände OK\_NOCARRY und OK\_END\_ERR sowie des Fehlers ERR\_NOADD beim Lösen einer Multiplikation mit Übertrag.

Rechnungsart, welche für Repräsentationszwecke für am besten geeignet befunden wurde. Zur strikten Trennung der zwei Statusmengen verwenden alle Bezeichner der OK-Zustandsmenge das Präfix `OK_`, jene der Fehlermenge das Präfix `ERR_`.

## 4.6 Datenbankdesign

Der Mathe-Multi-Trainer setzt zur Speicherung von wichtigen und hilfreichen Datenwerten auf die Verwendung einer relationalen MySQL-Datenbank. Die automatische Verwaltung und Sicherung von Lerndaten ist speziell für die Leistungsbewertung der Schüler von enormer Wichtigkeit. Lehrkräften wird durch die Statistikauswertung (siehe nächster Abschnitt 4.7) fast voller Zugriff auf alle leistungsrelevanten Informationen ihrer Schüler gegeben. Dadurch ist es möglich, aufschlussreiche Lernanalysen für jede Benutzerin und jeden Benutzer zu erstellen, individuelle Lernfortschritte zu erkennen und, wenn nötig, geeignete pädagogische Maßnahmen zur Leistungsbeeinflussung der Schüler zu setzen. Demgegenüber nützt der Generierprozess für Multiplikationen ebenso die Datenbank als Informationsgeber, indem der entwickelte Algorithmus zur Ermittlung der Beispiele für bestimmte Aufgabenstellungen gewisse Faktoren aus der Datenbank bezieht. In Abbildung 4.12 ist der Datenbank-aufbau der Lernapplikation nachgezeichnet. Es sind fünf Tabellen definiert, welche anschließend näher erläutert werden.

### 4.6.1 Tabelle `multipliers`

Diese Datenbanktabelle hat die Speicherung der zwei- und dreistelligen Multiplikatoren für die übertragslose Aufgabengenerierung inne, welche in Unterkapitel 4.4.1 vorgestellt wurde. Die Sicherung der Faktoren in der Datenbank hat den Vorteil einer problemlosen und konsistenten Datenbeschaffung durch den Algorithmus. Bei Nutzung der Datenschnittstelle sind zudem keine Einbußen in der Performanz zu erwarten. Die Tabelle `multipliers` enthält neben dem Primärschlüssel `multiplierId` lediglich Länge (`length`) und Wert des Multiplikators (`multiplier`). Die aktuellen Tabellenwerte der Faktoren sind in Aufstellungen 4.9 und 4.10 gelistet. Das Längenattribut nimmt dementsprechend nur die Werte 2 und 3 an. Die Wahrscheinlichkeitsverteilungen der verschiedenen Faktoren ergeben sich anhand der variablen Speicheranzahl eines definierten Wertes in der Tabelle. Eine Adaption oder Konfiguration der verwendeten zwei- beziehungsweise dreistelligen Multiplikatoren kann auf Wunsch durch Administratoren im Backend der Lernapplikation vorgenommen werden.

### 4.6.2 Tabelle `answers`

Die Antworttabelle ist für die Sicherung von Überblicksdaten zu jeder einzelnen Aufgabenstellung verantwortlich. Der höchste Wert des gespeicherten Primärschlüssels `answerId` verrät dabei die Gesamtanzahl der vom Programm generierten Aufgabenstellungen für alle Anwender der Lernapplikation. Die Zuordnung von Schülern zu von ihnen getätigten Lösungsantworten von Multiplikationen wird durch das Tabellenattribut `userId`, also der eindeutigen Kennzeichnung einer Anwenderin oder eines Anwenders, erreicht. Zusätzlich wird die für jede Schülerin beziehungsweise für jeden Schüler individuelle Gesamtanzahl von Beantwortungen in der Tabelle festgehalten (`userAnswerId`).

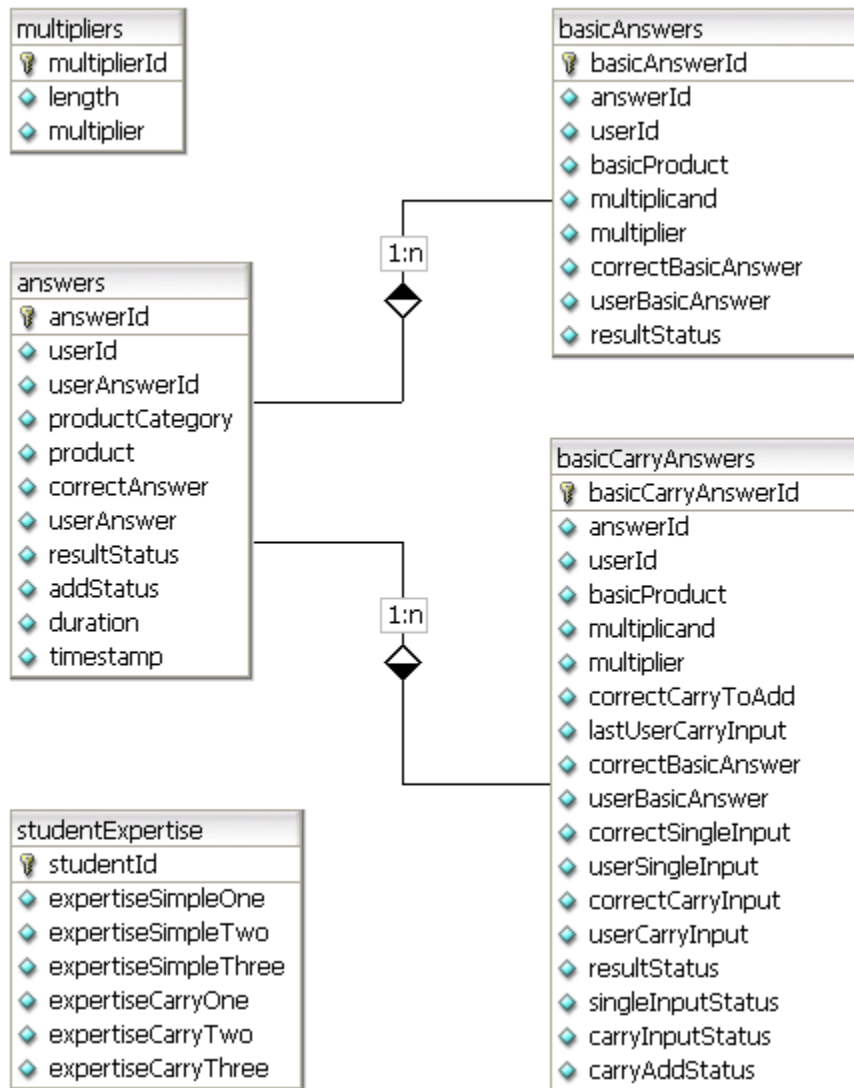


Abbildung 4.12: Das Datenbankmodell des Mathe-Multi-Trainers.



Die jeweils vom Lerntrainer generierte Aufgabenstellung findet sich im Element `product` wieder, in der Form *Multiplikand\*Multiplikator*. Das Attribut `productCategory` kennzeichnet die Problemgruppe der ermittelten Multiplikation. Dessen Bezeichner und Wertebereich sind in Aufstellung 4.15 dargestellt. Die einzig mögliche korrekte Gesamtlösung einer Aufgabenstellung repräsentiert das Attribut `correctAnswer`, während die tatsächliche Benutzerantwort im Tabellenelement `userAnswer` gespeichert wird. Der vom Evaluierungsalgorithmus berechnete allgemeine Ergebnisstatus (`resultStatus`) wird ebenso gesichert wie die Dauer der Lösungsfindung (`duration`) und der Zeitstempel beim Absenden aller Eingaben (`timestamp`). Die Lösungsdauer einer Beantwortung erstreckt sich vom Zeitpunkt der Anzeige der Multiplikation im Browserfenster bis zum Betätigen des Antwortbuttons. Da für gewisse Problemstellungen eine Gesamtaddition von zeilenweisen Teilprodukten notwendig ist, beinhaltet die Tabelle ebenso den Ergebnisstatus einer solchen Addition (`addStatus`). Sollte eine Multiplikation keine Addition verlangen – dieser Fall ist bei einstelligen Multiplikatoren gegeben – so enthält `addStatus` den Wert `OFF`.

<b>productCategory</b>	<b>Bedeutung</b>
2x1	Multiplikation der Problemgruppe 1 ohne Übertrag
2x1c	Multiplikation der Problemgruppe 1 mit Übertrag
3x1	Multiplikation der Problemgruppe 2 ohne Übertrag
3x1c	Multiplikation der Problemgruppe 2 mit Übertrag
4x1	Multiplikation der Problemgruppe 3 ohne Übertrag
4x1c	Multiplikation der Problemgruppe 3 mit Übertrag
2x2	Multiplikation der Problemgruppe 4 ohne Übertrag
2x2c	Multiplikation der Problemgruppe 4 mit Übertrag
3x2	Multiplikation der Problemgruppe 5 ohne Übertrag
3x2c	Multiplikation der Problemgruppe 5 mit Übertrag
4x2	Multiplikation der Problemgruppe 6 ohne Übertrag
4x2c	Multiplikation der Problemgruppe 6 mit Übertrag
3x3	Multiplikation der Problemgruppe 7 ohne Übertrag
3x3c	Multiplikation der Problemgruppe 7 mit Übertrag
4x3	Multiplikation der Problemgruppe 8 ohne Übertrag
4x3c	Multiplikation der Problemgruppe 8 mit Übertrag

Tabelle 4.15: Alle möglichen Werte des Attributs `productCategory` der Antworttabelle einschließlich entsprechender Erklärungen.

### 4.6.3 Tabelle `basicAnswers`

Die Datenbanktabelle `basicAnswers` ist eng verknüpft mit der Antworttabelle `answers`, welche im vorhergehenden Abschnitt beschrieben wurde. Sie enthält detaillierte Daten zu allen generierten Aufgabenstellungen ohne Übertrag. Genauer gesagt enthält die Tabelle Angaben zu allen zergliederten Teilprodukten einer Multiplikation, welche selbst durch einzelne Eingabezellen repräsentiert werden. Die Verbindung zur Tabelle `answers` wird durch die zwei identen Attribute `userId` und `answerId` der beiden Tabellen hergestellt. Eine Problemstellung spaltet sich immer in mehrere Teilmultiplikationen auf, deren Ergebnisprodukte die einzelnen Antwortzellen einer Zeile wiedergeben. Diese Eigenschaft spiegelt sich in der 1:n-Beziehung der beiden Tabellen wider. Eine Teilmultiplikation (`basicProduct`) ergibt sich durch jeweils einstellige Faktoren, deren Bezeichner in der Tabelle durch die Attribute `multiplicand` und `multiplier` abgebildet sind. Die korrekte Soll-Lösung (`correctBasicAnswer`) und die von Benutzern tatsächlich in das entsprechende Antwortfeld eingetragene Ergebniszahl (`userBasicAnswer`) wird ebenso gespeichert wie der vom Auswertungsalgorithmus automatisch generierte Ergebnisstatus `resultStatus`.

### 4.6.4 Tabelle `basicCarryAnswers`

Synonym zur Tabelle für Detailantworten von übertragslosen Problemstellungen ist die Tabelle `basicCarryAnswers` für Eingaben von übertragsbehafteten Multiplikationsrechnungen zuständig. Diese ist durch die gemeinsamen Attribute `answerId` und `userId` in gleicher Weise mit der Antworttabelle `answers` verknüpft wie die Tabelle `basicAnswers`. Die Ergebnisse der Teilmultiplikationen einer solchen Rechnung sind nun durch die Übertragseigenschaft in zwei Eingabefelder aufgeteilt – die Einerzelle und das Übertragselement. Dadurch ergibt sich eine Reihe von zusätzlichen Attributen für die Tabelle. In derselben Weise wie zuvor für die Tabelle `basicAnswers` wird ein Teilprodukt (`basicProduct`) mit dessen einstelligen Faktoren Multiplikand (`multiplicand`) und Multiplikator (`multiplier`) in der übertragsbehafteten Detailtabelle gespeichert, ebenso der korrekte (`correctBasicAnswer`) sowie der individuell eingegebene (`userBasicAnswer`) zweistellige Lösungswert für eine Teilmultiplikation. Dazu liefern die Attribute `correctCarryToAdd` und `lastUserCarryInput` Informationen zu richtigen und von Benutzern tatsächlich eingetragenen Addierüberträgen, welche einem Teilergebnis aufsummiert werden, sofern vorhanden. Das erste äußerst rechte Teilprodukt einer Rechnungszeile ergibt sich beispielsweise immer nur aus einer reinen Multiplikation, die beiden erwähnten Übertragsattribute der Tabelle sind für diese Teilantwort nicht relevant und daher stets durch den Wert `OFF` gekennzeichnet. Durch die Präsenz von nunmehr zwei Eingabezellen werden als zusätzliche Informationen zu den erwähnten Antwortattributen auch die jeweils korrekten und individuellen Usereingaben zu Einer- und Übertragsfeldern mitgespeichert, das sind die Elemente `correctSingleInput` und `userSingleInput` beziehungsweise `correctCarryInput` und `userCarryInput`. Für äußerst linke Teilprodukte einer Ergebniszeile sind diese Attribute aufgrund des Fehlens einer Übertragszelle immer ohne Information (`OFF`). Der berechnete Ergebnisstatus einer Teilrechnung findet sich im Tabellenattribut `resultStatus` wieder. Als Zusatzinformationen werden die jeweiligen Antwortzustände für die Eingaben in Einerfeld (`singleInputStatus`) und Übertragszelle (`carryInputStatus`) dargestellt, sowie der Additionsstatus für den Übertrag (`carryAddStatus`). Aus diesen partiellen Zuständen wird für gewöhnlich der Ergebnisstatus reproduziert. Abermals werden diese Attribute für bestimmte

Teilrechnungen situationsbedingt auf den Wert OFF gestellt.

#### 4.6.5 Tabelle `studentExpertise`

Die Datenbanktabelle `studentExpertise` ist für den Trainingsmodus der Lernapplikation maßgebend. Sie enthält den aktuellen Übungsfortschritt von Schülern, dargestellt durch individuelle Kompetenzstufen. Durch die eindeutige Identifikationsnummer einer Schülerin beziehungsweise eines Schülers (`studentId`) werden diese spezifischen Kompetenzkategorien ebendiesen Benutzern zugewiesen. Die Tabellenattribute `expertiseSimpleOne`, `expertiseSimpleTwo` und `expertiseSimpleThree` kennzeichnen den Lernstatus für übertragslose Aufgabenstellungen mit ein-, zwei- und dreistelligen Multiplikatoren. In gleicher Weise werden aktuelle Kompetenzen für Beispiele mit Übertrag mit schrittweiser Stellenzunahme des Multiplikators anhand der Tabellenelemente `expertiseCarryOne`, `expertiseCarryTwo` und `expertiseCarryThree` abgebildet. Die Interpretation der jeweiligen Attributwerte der Kompetenzelemente ist Sache des verwendeten Trainings-Algorithmus. Dieser wird ausführlich in Abschnitt 4.8 erklärt, wobei die vorliegende Implementation des Mathe-Multi-Trainers höhere Kompetenzwerte als gesteigerte Lernfähigkeit auslegt.

## 4.7 Statistikauswertung

Eine für Lehrkräfte äußerst wichtige Komponente des Mathe-Multi-Trainers umfasst die Möglichkeit der Anzeige von schülereigenen Leistungsauswertungen. Dabei können Übersichtstabellen mit allgemeinen Informationen zu anwenderspezifischen Interaktionen mit der Lernapplikation bis hin zu detaillierten Aufstellungen von einzelnen Antworteingaben der Schüler ausgewählt werden. Die Statistikauswertung ist nur für registrierte Lehrpersonen durch den Link *Statistiken* erreichbar. Intern arbeitet der Statistik-Controller der Implementierung eng mit den Datenbanktabellen `answers`, `basicAnswers` und `basicCarryAnswers` zusammen. Aus den genannten Tabellen bezieht die Implementierung ausgesuchte Daten zur Anzeige. Bei Aufruf des Statistik-Links durch eine Lehrkraft wird anfangs eine universelle Überblicksseite ausgegeben. Diese ist in Abbildung 4.13 dargestellt und enthält Informationen zu allgemeinen Lernaktivitäten mit dem Mathe-Multi-Trainer durch Schüler, welche dieser Lehrperson zugeordnet sind. Anders ausgedrückt bedeutet die Zuweisung zu einer Lehrkraft beziehungsweise schlichtweg die Auflistung von Schülern in der Übersichtsseite der Statistikauswertung, dass diese von der aufrufenden Lehrperson unterrichtet werden. Schüler sind zudem in ihre jeweiligen Klassen unterteilt dargestellt. Die Beispielabbildung zeigt zwei registrierte Schulklassen, von denen ebenso registrierte Schüler mit dem Lerntrainer interagiert haben. In diesem Fall handelt es sich, wie ersichtlich, um die *Klasse 2a* mit den Schülern *Michael User* und *Max Musterschüler*, sowie die *Klasse Testarium*, der die Schüler *Betty Langsam*, *Paul Schüler* und *Melanie Test* zugeordnet sind. Es sei angemerkt, dass die Registrierung und Zuordnung von Schulklassen und Schülern im Usermanagement des externen Lernservers gehandhabt wird.

**Mathe-Multi-Trainer**

» Logout  
» Üben  
» Statistiken

### Nutzerstatistik

**Klasse 2a**

Benutzername	Beispiele gesamt	Korrekt gelöst	Fehlerhaft gelöst	Erfolgsquote	Fehlerhäufung	Details
Michael User	105	39	66	37.14 %	Diverse Fehler	» HIER «
Max Musterschüler	21	15	6	71.43 %	Diverse Fehler	» HIER «

**Klasse Testarium**

Benutzername	Beispiele gesamt	Korrekt gelöst	Fehlerhaft gelöst	Erfolgsquote	Fehlerhäufung	Details
Betty Langsam	11	3	8	27.27 %	Diverse Fehler	» HIER «
Paul Schüler	32	30	2	93.75 %	1x1 Fehler	» HIER «
Melanie Test	8	4	4	50 %	Eingabefehler	» HIER «

© 2012

Abbildung 4.13: Beispielansicht der Übersichtsseite der Statistikauswertung des Mathe-Multi-Trainers mit zwei registrierten Klassen.

Die Übersichtsstatistik aus Darstellung 4.13 enthält für jede Schülerin und jeden Schüler individuelle Angaben zur Gesamtanzahl von beantworteten Multiplikationsaufgaben, der Menge an korrekt und fehlerhaft gelösten Beispielen und der daraus berechneten generellen Erfolgsquote, die in Prozentwerten angegeben wird. Dessen Tabellenzelle deutet Lernerfolg durch eine grüne Hintergrundfarbe an, wenn die Erfolgsquote einen Wert größer als 70 Prozent ausmacht. Liegt sie darunter, wird das jeweilige Tabellenelement rot eingefärbt. Dadurch ist mit einem Blick erkennbar, welche Schüler erfolgreich beziehungsweise fehlerhaft arbeiten. Die Spalte *Fehlerhäufung* gibt an, ob ein bestimmter Fehler bei der Lösungsfindung aller Aufgaben regelmäßig aufgetreten ist oder ob unterschiedliche Fehlerarten vorgekommen sind. Im erstgenannten Fall wird die Bedeutung des genauen Fehlerstatus ausgegeben, in Letzterem wird die Information *Diverse Fehler* angezeigt. Beträgt die Erfolgsquote 100 Prozent, wird folgerichtig *Kein Fehler* im Fehlerhäufungs-Reiter zurückgegeben. Das Tabellenelement *Details* enthält Links zu den jeweiligen Detailstatistiken für eine bestimmte Schülerin beziehungsweise für einen Schüler.

Ein Aufruf einer derartigen Detailaufstellung führt zu einer weiteren Übersichtstabelle, welche nun für einen definierten User angezeigt wird. Dessen Inhalt ist in Antwortdaten von Aufgabenstellungen mit und von jenen ohne Übertrag unterteilt. Es sind dieselben Tabellenelemente wie zuvor in der allgemeinen Übersichtsstatistik ausgeführt. Der Link im Reiter *Details* belegt eine ausgewählte Zeile bei Selektion mit grauem Hintergrund und führt zu einer weiteren Untertabelle, welche genauere Angaben zu den Beantwortungen der Beispielkategorie der gewählten Zeile auflistet. Angesprochenes Szenario wurde bereits in Abbildung 4.5 erklärt und dargestellt. Es zeigt die Detailstatistik für den Benutzer *Max Musterschüler* mit Auswahl der Problemkategorie *OHNE Übertrag*. Beide Aufstellungen sind folglich mit den bis dahin gespeicherten Lernaktivitäten des Schülers befüllt. Führt man die Auflistung aller Detailtabellen durch mehrmaliges Betätigen weiterführender Detaillinks fort, so ergibt sich als letzte Untertabelle eine Auflistung aller einzelnen Eingaben der Antwortfelder für eine Multiplikation. Beispielhaft dafür ist Abbildung 4.14, welche das vorhergehende Statistikszenario für den Schüler *Max Musterschüler* ausmalt und alle Detailtabellen einschließlich jener für die übertragslose Multiplikation  $82 * 43$  zeigt. Nun sind absolut alle Benutzereingaben im Detail und nachvollziehbar dargestellt. Die dritte Abstufungstabelle enthält Angaben zu Beantwortungen von Beispielen für bestimmte Problemgruppen. Neben den tatsächlichen Rechnungsbeispielen, der korrekten und benutzerspezifischen Lösungsantworten und dem daraus berechneten Ergebnisstatus sind zusätzlich die Lösungsdauer und der Zeitstempel der Beantwortung angegeben. Diese Informationen können für Lehrkräfte wichtige Hinweise für eine positive wie negative Leistungsbemessung liefern. Der Ergebnisstatus hält nicht den tatsächlichen Ergebniscode, sondern immer die Kurzbeschreibung dessen (nach Tabellen 4.13 und 4.14). Schlussendlich wird als letzte Aufstellung eine selektierte Aufgabenstellung mit all ihren Benutzereingaben und den einzelnen Ergebniszuständen für alle Teilrechnungen inklusive einer möglichen Gesamtaddition gezeigt. Die in Darstellung 4.14 selektierte Multiplikation einschließlich der individuellen Eingaben und Ergebniszustände sind Ergebnis der Lösungsfindung, welche bereits in Abbildung 4.8 bei der Behandlung von Auswertungen für übertragslose Multiplikationen vorgestellt wurde. Wie ersichtlich, werden OK-Zustände, Vertauschungs- und Additionsfehler den jeweiligen Teilantworten exakt zugeordnet. Obwohl das richtige Endresultat in die entsprechenden Felder eingetragen wurde, ist die Lösungsfindung aufgrund der vorliegenden Fehler nicht korrekt und stellt einen Sonderfall dar, welcher vom Programm erkannt wird.

## Mathe-Multi-Trainer

- » Logout
- » Üben
- » Statistiken

### Detailstatistik für Max Musterschüler

Problemkategorie	Anzahl	Korrekt gelöst	Fehlerhaft gelöst	Erfolgsquote	Fehlerhäufung	Details
OHNE Übertrag	10	8	2	80 %	Diverse Fehler	<a href="#">HIER</a>
MIT Übertrag	11	7	4	63.64 %	Diverse Fehler	<a href="#">HIER</a>

OHNE Übertrag

Problemgruppe	Anzahl	Korrekt gelöst	Fehlerhaft gelöst	Erfolgsquote	Fehlerhäufung	Details
2x1, 3x1, 4x1	5	4	1	80 %	1x1 Fehler	<a href="#">HIER</a>
2x2, 3x2, 4x2	3	2	1	66.67 %	Diverse Fehler	<a href="#">HIER</a>
3x3, 4x3	2	2	0	100 %	Kein Fehler	<a href="#">HIER</a>

Gruppe 2x2, 3x2, 4x2 OHNE Übertrag

Problem ID	Multiplikation	Korrekte Lösung	Benutzer-Antwort	Ergebnisstatus	Lösungsdauer	Zeitstempel	Details
9	421*23	9683	9683	OK	00:00:17	2012-07-19 19:53:31	<a href="#">HIER</a>
12	82*43	3526	3526	Diverse Fehler	00:00:23	2012-07-20 19:50:07	<a href="#">HIER</a>
13	5321*23	122383	122383	OK	00:00:25	2012-07-26 20:52:53	<a href="#">HIER</a>

nach oben ↑

82\*43

Teil-Multiplikation	Korrekte Lösung	Benutzer-Antwort	Ergebnisstatus
2*4	8	8	OK
8*4	32	23	Eintragungsfehler (Vertauschung)
2*3	6	6	OK
8*3	24	42	Eintragungsfehler (Vertauschung)

Gesamtaddition

Korrekte Lösung	Benutzer-Antwort	Ergebnisstatus
3526	3526	Additionsfehler (trotz korrekter Antwort)

nach oben ↑

© 2012

Abbildung 4.14: Die Detailstatistik für den Benutzer *Max Musterschüler* bei Auswahl aller möglichen Untertabellen für eine bestimmte Aufgabenstellung.

In ähnlicher Weise wie zuvor umfasst der Ausschnitt einer Untertabelle aus Abbildung 4.15 die detaillierte Statistik über eine Multiplikation mit Übertrag, welche bereits in Darstellung 4.10 behandelt wurde. Die dort beschriebenen Ergebniszustände `OK_ERR`, `OK_END_ERR`, `ERR_REV_IN`, `ERR_1x1` und `ERR_ADD` finden sich nun als Kurzbeschreibungen im Reiter *Ergebnisstatus* wieder. Zusätzlich sind für übertragsbehaftete Aufgabenstellungen neben den korrekten und benutzerspezifischen Teilantworten auch informative Zusatzangaben wie korrekt und tatsächlich (eingabeabhängig) zu addierende Überträge sowie neue Addierüberträge für nachfolgende Teilmultiplikationen dargestellt. Sind für gewisse Rechnungen keine Überträgeinträge möglich, wird der Status `OFF` zurückgegeben.

387*64						
Teil-Multiplikation	Addier-Übertrag (korrekt)	Korrekte Lösung	Addier-Übertrag (Benutzer-eingabe)	Benutzer-Antwort	Neuer Übertrag	Ergebnisstatus
7*6	OFF	42	OFF	24	2	Eintragungsfehler (Vertauschung)
8*6	4	52	2	50	5	OK (inkludiert Folgefehler)
3*6	5	23	5	23	OFF	OK
7*4	OFF	28	OFF	38	3	1x1 Fehler
8*4	2	34	3	34	3	Übertrag-Addierfehler (trotz korrekter Antwort)
3*4	3	15	3	15	OFF	OK

Gesamtaddition		
Korrekte Lösung	Benutzer-Antwort	Ergebnisstatus
24768	24588	Addition OK (inkludiert Folgefehler)

nach oben ↑

Abbildung 4.15: Ausschnitt aus der Detailaufstellung für eine übertragsbehaftete Multiplikation.

## 4.8 Programmablauf im Trainingsmodus

Das Herzstück des Mathe-Multi-Trainers bildet der adaptive Trainingsmodus, der automatisch nach dem Anmeldevorgang gestartet wird und anderenfalls über den Link *Üben* erreichbar ist. Dieser setzt sich aus allen wesentlichen Funktionen und Berechnungsvorschriften des Programms zusammen, wie den Algorithmen zur Generierung und Evaluierung der Problemstellungen und der Speicherung aller relevanten Werte in entsprechende Datenbanktabellen bei der Antwortgebung. Der Trainingsmodus ist derjenige Teil der Lernapplikation, welcher Schüler beim Üben und Erlernen der mehrstelligen Multiplikation durch dessen adaptive Beispielgebung Unterstützung leistet – abhängig von individuellem Können werden angepasste Aufgabenstellungen für Schüler ausgegeben. Um zu jeder Zeit Kenntnis über die Leistungsfähigkeit von Benutzern zu haben, werden anwenderspezifische Kompetenzen für bestimmte Problemgruppen von Multiplikationen nach jeder Beantwortung in der Datenbanktabelle `studentExpertise` (siehe Unterabschnitt 4.6.5) aktualisiert gespeichert. Die Tabelle enthält jeweils drei unterschiedliche Kompetenzstufen für Multiplikationen mit Übertrag und für diejenigen ohne Übertragsrechnung. Diese drei Kompetenzkategorien sind ident mit jenen aus Tabelle 4.1 und umfassen alle acht Problemgruppen. Für den entwickelten Trainingsalgorithmus relevant sind genau

diese drei erwähnten Leistungsstufen, zusätzlich zur Unterteilung in übertragslose und übertragsbehaftete Problemstellungen. Durch letztere Differenzierung erhalten beide Beispielgruppen für deren Ermittlung von der Berechnungsvorschrift dieselbe Auftrittswahrscheinlichkeit. Das bedeutet, dass der Trainingsalgorithmus willkürlich – jedoch gleichwahrscheinlichkeitsverteilt – entscheidet, ob ein Beispiel mit oder ohne Übertrag zurückgeliefert wird, unabhängig von dessen Kompetenzkategorie. Dies soll zu einer universellen Erlernung der mehrstelligen Multiplikation beitragen.

Bei erstmaliger Anwendung des Übungsmodus erhalten Schüler mit einer Wahrscheinlichkeit von 90 Prozent eine Aufgabenstellung der einfachsten Kompetenzkategorie, welche nur Probleme mit einstelligen Multiplikatoren enthält. Zu gleicher Wahrscheinlichkeit ist dieses Beispiel übertragsbehaftet oder ohne Ausführung der Übertragsrechnung. Der Pool an möglichen Aufgabenstellungen für eine Kompetenzkategorie erstreckt sich über mehrere Problemgruppen, welche ebenfalls per Zufall ausgewählt werden. Es kann zum Beispiel der Fall eintreten, dass das erste zur Anzeige gebrachte Beispiel für eine Übende oder einen Übenden eine Multiplikation mit Übertrag der Problemgruppe 3 (vierstelliger Multiplikand) darstellt. Dies stellt die formal schwierigste Beispielkategorie für die erste Kompetenzstufe dar. Auf den ersten Blick könnte dieses Szenario als zu kompliziert erachtet werden. Dennoch wird diese Vorgehensweise einer *einfacheren* Abhandlung durch weitere Einschränkungen auf beispielsweise ausschließlich Problemgruppen vorgezogen. Der Grund liegt in einer breitgefächerten, aber doch eingeschränkten, Beispielgebung, welche genau eine Kompetenzkategorie mit einer Fülle von unterschiedlichen Multiplikationen abdeckt und Schülern damit die Möglichkeit der Lösung von variablen und nicht starren Aufgabenstellungen gibt.

Erfolgt nun eine fehlerfreie Antwort der dargestellten Problemstellung für die erste Kompetenzstufe, so wird das entsprechende Attribut `expertiseSimpleOne` oder `expertiseCarryOne`, je nach Übertragsausführung, in der Tabelle `studentExpertise` um den Faktor 1 erhöht. Tritt zumindest ein Berechnungsfehler auf, erfolgt in diesem Fall keine Erhöhung des entsprechenden Elements. Für bereits positive Attributwerte erfolgt eine Erniedrigung des Datenbankelements um den Wert 1. Ein weiteres Mal sei erwähnt, dass im Frontend positive Rückmeldungen von Lösungsfindungen nur das Endergebnis umfassen und *innere* Rechnungsfehler nicht mit einschließen. Dieser Fall ist erwartungsgemäß relativ unwahrscheinlich, dennoch behandelt das Backend intern absolut jeden ermittelten Fehlerstatus als negative Gesamtlösung.

Ein Beispiel der ersten Kompetenzstufe wird nun solange erzeugt, bis das entsprechende Datenbankattribut den Wert 3 aufweist. Erst dann erfolgt für die Generierung der Multiplikationen eine Erhöhung der Kompetenzkategorie auf die zweite Stufe. Nun ist es möglich, dass zum Beispiel das Element für einfache übertragslose Multiplikationen bereits den Wert 3 aufweist, dasjenige für Probleme mit Übertrag derselben Kategorie hingegen noch nicht. Der Trainingsalgorithmus behandelt diesen Fall in der Art und Weise, dass bei Auftreten eines übertragsbehafteten Beispiels eben noch die erste Kompetenzklasse mit einstelligen Multiplikatoren gewählt wird, für Aufgaben ohne Übertrag indes Beispiele mit bereits zweistelligen Multiplikatoren erzeugt werden. Die Berechnungsvorschrift arbeitet also unabhängig eines etwaigen Übertragvorkommens. Für die zweite und dritte Kompetenzkategorie erfolgt dieselbe Behandlung der entsprechenden Elemente in der Datenbank wie dies zuvor für Vermögensstufe 1 ausgeführt wurde. Bei erfolgreichen fehlerlosen Beantwortungen von Multiplikationen wird das betreffende Kompetenzattribut um den Wert 1 erhöht, bei fehlerhafter Lösung um den Wert 1 erniedrigt. Ein Attribut umfasst einen Definitionsbereich von 0 (kein Lerneffekt) bis 3 (erfolgreicher Lerneffekt), kann daher nie negative Werte oder Faktoren größer als 3 annehmen. Bei Erreichen des Werts 3 für eine Kompetenzkategorie erfolgt automatisch die Auswahl der nächsthöheren Stufe für die



daraufliegende Beispielgenerierung. Sind alle Attributwerte mit dem Faktor 3 belegt, wird ein allgemeines Übungsbeispiel erzeugt, welches im nächsten Absatz näher erklärt wird. Diese Vorgehensweise des Trainingsalgorithmus gewährleistet, dass Schüler bei positivem Lernfortschritt stetig schwierigere Beispiele zu lösen haben. Dies trägt zur schrittweisen Erlernung der mehrstelligen Multiplikation wesentlich bei.

Es wurde bereits erwähnt, dass Aufgabenstellungen für passende Kompetenzkategorien immer mit einer Wahrscheinlichkeit von 90 Prozent generiert werden. Zu den restlichen zehn Prozent erfolgt die Ermittlung eines sogenannten allgemeinen Übungsbeispiels. Dieses wird aus der Beispielmenge von bereits gelösten Kompetenzstufen in zufälliger Weise erzeugt, das heißt für Kategorien, deren entsprechende Attribute in der Tabelle `studentExpertise` den Wert 3 aufweisen. Ist dies noch nicht der Fall, wird ein Beispiel aus jenen Kategorien ermittelt, welche bereits erste positive Lösungsfindungen beinhalten, gekennzeichnet durch die Attributwerte 1 oder 2. Tritt der Fall ein, dass ein allgemeines Übungsbeispiel bei gegenwärtigem Nichtvorhandensein von positiven Kompetenzwerten in der Tabelle generiert werden soll, so wird eine Multiplikation der einfachsten Problemkategorie erstellt, zu gleicher Wahrscheinlichkeit mit oder ohne Übertrag. Wurden alle Kompetenzstufen positiv erledigt – alle entsprechenden Datenbankattribute beinhalten den Faktor 3 – so wird automatisch ein zufälliges Trainingsproblem aus allen möglichen Kompetenzbereichen ausgewählt. Wird ein allgemeines Übungsbeispiel zur Anzeige gebracht und fehlerhaft gelöst, erfolgt für die betreffende Kompetenzstufe des Beispiels eine Erniedrigung des entsprechenden Wertes in der Datenbank. Diese Rückstufung wird vom Algorithmus im *gewöhnlichen* Übungsmodus bei der Bereitstellung der daran anschließenden Multiplikation berücksichtigt.

Abbildung 4.16 zeigt das Ablaufschema des entwickelten Trainingsalgorithmus. Nach Auswahl des Übungsmodus erfolgt zu Beginn ein Check, ob der betreffende User bereits in der Kompetenztabelle der Datenbank registriert ist, oder ob es sich um den erstmaligen Aufruf des Übungsmodus handelt. In ersterem Fall werden die entsprechenden Werte aus der Datenbank der Anwenderin oder dem Anwender zugeordnet. Anderenfalls erfolgt eine Registrierung des Users mit initialen Kompetenzwerten von 0 für die Tabellenattribute. Anschließend wird eine Zufallszahl geworfen und in Prozentwerten abgebildet. Anhand der zuvor erklärten Auftrittswahrscheinlichkeiten für Aufgabenstellungen mit und ohne Übertrag (gleichverteilt) und für die Ermittlung eines Übungsbeispiels gibt die Zufallszahl den weiteren Ablauf wie ersichtlich vor. Für die Erzeugung der Problemstellungen, ausgenommen sind jene des allgemeinen Übungsbeispiels (zehnprozentige Auftrittswahrscheinlichkeit), wird vorab geprüft, welche Kompetenzkategorien bereits gelöst wurden, um die zu generierenden Multiplikationen dem aktuellen Lernstand der Schüler anzupassen. Nach Erzeugung und Anzeige des Beispiels im Browserfenster wird auf die Benutzerantwort gewartet. Sobald die Lösungsfindung abgeschickt wurde, erfolgt die Auswertung der Ergebniszustände, um danach die entsprechenden Kompetenzattribute für die gegenwärtige Schülerin beziehungsweise den Schüler in der Datenbank zu aktualisieren.

Der Vorgang von der Prüfung aller Kompetenzelemente auf den Wert 3 bis zur Erzeugung der Multiplikationen ist hier generalisierend als Kompetenzmodul *KM* zusammengefasst. Die genaue Abfolge dieses Moduls ist in Abbildung 4.17 gegeben. Dabei werden die einzelnen Kompetenzkategorien K1, K2 und K3 je nach Problemgruppe (mit oder ohne Übertrag) durchgehend auf Werte kleiner als 3 verglichen, um bei positiver Rückmeldung die betreffende Multiplikation zu generieren. Sind alle einfachen übertraglosen Problemgruppen bereits positiv absolviert – das heißt deren entsprechenden Datenbankwerte weisen den Wert 3 auf – so wird geprüft, ob noch passende Kompetenzklassen mit Übertrag zu lösen sind. Ist auch dem nicht so, wird ein allgemeines Übungsbeispiel erzeugt.

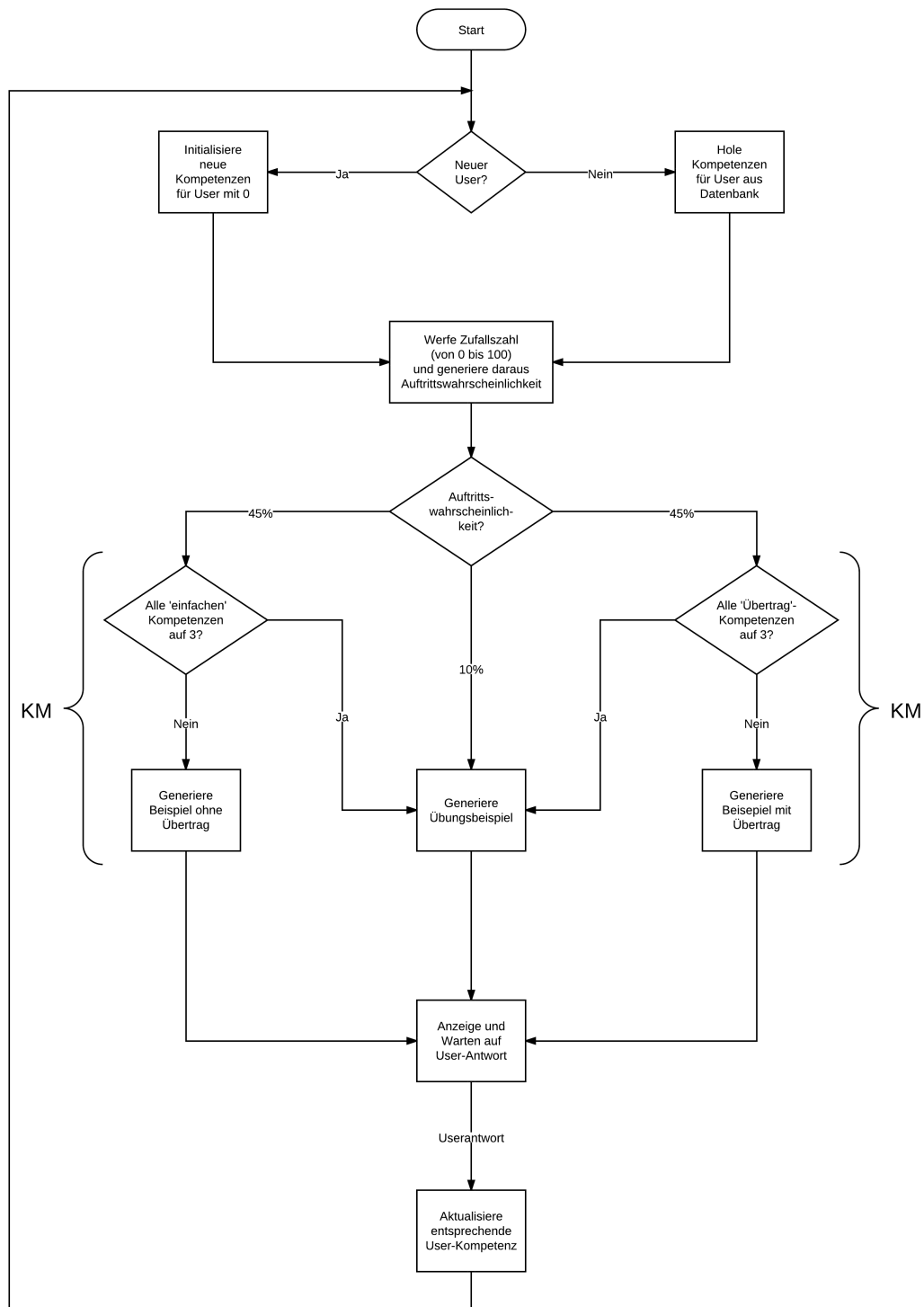


Abbildung 4.16: Programmablaufplan des Trainingsmodus des Mathe-Multi-Trainers.

Sind andererseits alle übertragsbehafteten Kategorien als gelöst markiert, wird ohne Prüfung der einfachen Kompetenzklassen direkt ein allgemeines Trainingsbeispiel generiert. Der Grund dafür liegt in der Annahme, dass die Lösung von übertragsbehafteten Aufgabenstellungen als komplexer und schwieriger als jene von übertragslosen Problemen angesehen wird.

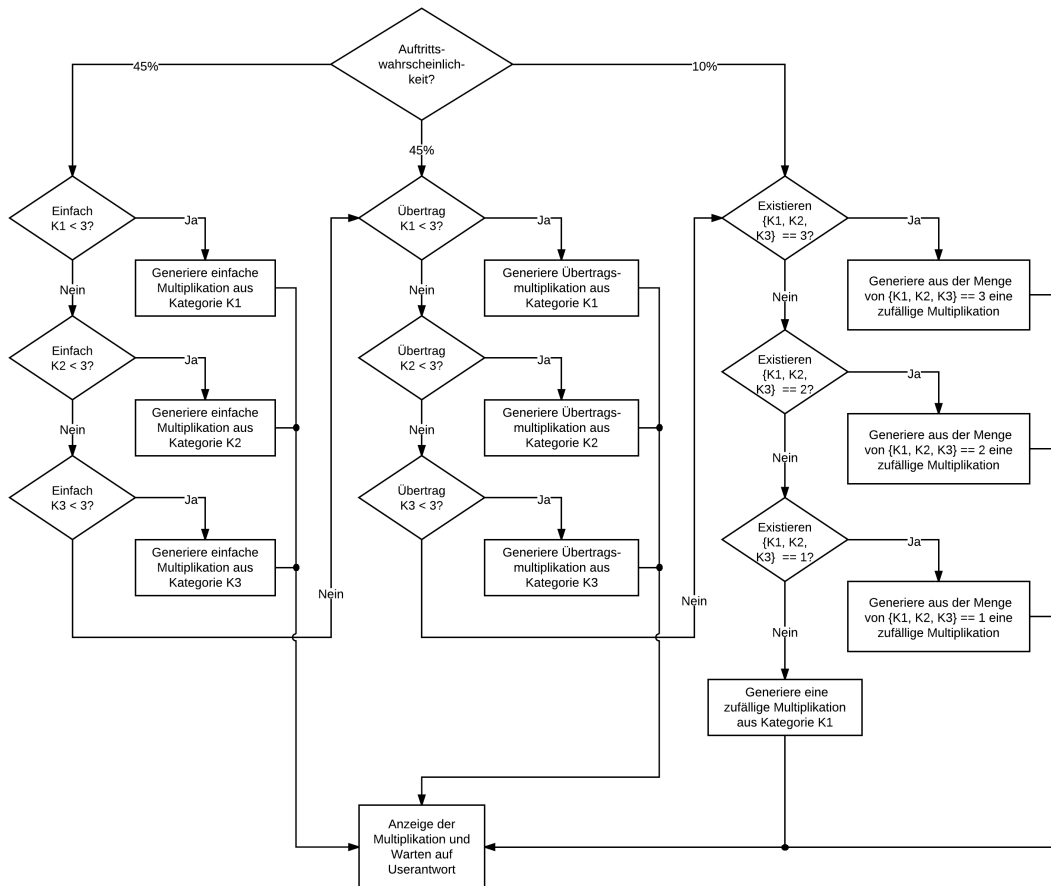


Abbildung 4.17: Das Kompetenzmodul des vorgestellten Trainingsalgorithmus.

Für Anwender besteht die Möglichkeit, ihren aktuellen Übungsfortschritt zurückzusetzen und wieder bei Null mit dem Trainingsmodus zu beginnen. Dazu dient der Funktionslink *Neustart* in der Navigationsleiste des Lernprogramms. Wird der Link betätigt, erscheint eine nochmalige Information inklusive eines Bestätigungsbuttons zur tatsächlichen Rückversetzung der gesamten Übungsdaten. Intern werden dabei alle Attribute der Kompetenztabelle auf den Wert 0 gesetzt. Die Möglichkeit des Resets lässt Benutzer eine erneute Überprüfung ihres Wissens beziehungsweise eine Wiederholung des gesamten Lerntrainings von Anfang an durchführen.

## 4.9 Bereitstellung von Webservices

Die Implementation des Mathe-Multi-Trainers stellt eine Web-Programmierschnittstelle zu dessen wesentlichen Funktionen für ausgewählte Anwendungsprogramme zur Verfügung. Dabei handelt es sich um den Trainingsalgorithmus, der intern auf die Algorithmen zur Generierung von Problemstellungen zugreift, und die Berechnungsvorschrift zur Evaluierung der Benutzerantworten einer mehrstelligen Multiplikation. Zur Zeit (August 2012) greifen Prototypen mobiler Lernapps für Smartphones und Tablet-Computer auf die Webservices des Mathe-Multi-Trainers zu. Die Entwicklungen dieser Anwendungen sind Bestandteil der E-Learning-Initiative an der Technischen Universität Graz und stehen noch in der Implementierungsphase. Durch die Möglichkeit des mobilen Zugriffs auf essenzielle Funktionen des Mathe-Multi-Trainers wird die in dieser Arbeit vorgestellte Erlernung der mehrstelligen Multiplikation mithilfe einer Lernanwendung ortsunabhängig. Die als Webservices bereitgestellten Funktionalitäten des Mathe-Multi-Trainers kommunizieren über das SOAP-Protokoll<sup>12</sup> und sind durch die Web Services Description Language (WSDL)<sup>13</sup> definiert.

---

<sup>12</sup> <http://www.w3.org/TR/soap/> [Zugriff am 21.08.2012]

<sup>13</sup> <http://www.w3.org/TR/wsd1> [Zugriff am 21.08.2012]

## 5 Schlussfolgerung und Ausblick

In der vorliegenden Arbeit wurde Learning Analytics nach Eigenheiten und Methoden hin untersucht, welche die Lernleistung von Schülern fördern. Gleichzeitig wurden Vorgehensweisen für einen verbesserten Unterricht gefunden. Learning Analytics befasst sich mit der Auffindung von Lernvorhersagen aus userspezifischen Daten, um die Leistung von Lernern positiv zu beeinflussen. Ein richtiger Eingriff bei Lernschwierigkeiten von Schülern wird dahingehend als wichtig erkannt. Als eine sowohl für Schüler wie auch für Lehrpersonen hilfreiche allgemeine Lernplattform wird ein sogenanntes Learning Management System aufgefasst. Es beinhaltet Lehrmaterialien für Ausbilder und Lernbehelfe für Schüler, zudem speichert und verwaltet die Plattform sämtliche Lernaktivitäten. Die Individualisierung von Lernplattformen führt zu Personal Learning Environments, welche strikt für spezielle Anwender ausgelegt werden und daher nur für diese im Lernbereich relevant sind. Ein Nachteil von derartigen Lernsystemen ist die meist zu ungenaue Definition von Lernaktivitäten. Die Soziale Netzwerkanalyse nimmt sich diesem Mangel an und interpretiert ein Online-Netzwerk als einen ungezwungenen Lernbereich. Durch die Vernetzung kann Wissen in Echtzeit ausgetauscht und angeeignet werden. Zudem wird versucht, alle Lernaktivitäten von Benutzern zu erfassen und gleichzeitig auszuwerten. Mithilfe eines Frameworks macht es Soziale Netzwerkanalyse möglich, wichtige Lerndaten in Visualisierungen umzuwandeln. Dadurch wird Lehrpersonen die Möglichkeit gegeben, schneller auf Lernmisserfolge von Schülern zu reagieren. Sogenannte Recommender-Systeme wurden als hilfreich bei der Behandlung von Ähnlichkeiten von Lerndaten gefunden, wodurch Empfehlungen zu brauchbaren Lerntechniken geliefert werden können. Zukünftige Entwicklungen von Lernsystemen sollten sich immer der Gefahr des Missbrauchs von sensiblen Daten bewusst sein, welche für diese Plattformen mitgespeichert werden. Ebenso sollte auf eine fehlerfreie Interpretation von Lerndaten geachtet werden.

Die Untersuchung von mathematischer Lernsoftware führte zum Ergebnis, dass die Verwendung von gegenwärtigen Anwendungen für Lehrende großteils zu schwierig und vielschichtig ist. Daher sollte das Hauptaugenmerk für zukünftige Implementationen auf der Zunahme von Intuitivität und Automatisierung liegen. Die Automatisierung gewisser Prozesse lässt Ausbilder auf jene Aufgaben konzentrieren, welche diese kompetenter als Computer zu lösen imstande sind. Beispielsweise kann ein System niemals die soziale Kompetenz von Lehrenden ersetzen. Knackpunkt in der Verwendung von Lernanwendungen im Unterricht ist die beidseitige Akzeptanz von Lehrpersonen und Software. Die größtmöglichen Vorteile einer Nutzung von derartigen Systemen zu herkömmlichen Unterrichtsmethoden liegen für Lehrkräfte in einer schnelleren und exakteren Ermittlung von schüler-eigenen Lernleistungen. Mathematische Lernanwendungen sollten immer nur als Unterstützer für Lehrkräfte auftreten, nie sollten diese lernrelevante Entscheidungen in Eigenregie treffen dürfen.

Die in dieser Arbeit vorgestellte Lernapplikation, der Mathe-Multi-Trainer, wurde nach vorher erwähnten Gesichtspunkten implementiert. Das Programm dient zur Unterstützung für Lehrkräfte, indem diesen wichtige Lerndaten als Statistikauswertungen zur Verfügung gestellt werden. Die Lernapplikation verwendet einen adaptiven Lernalgorithmus, der Schüler individuell nach deren Leistungsvermögen

einstuft und dazu passende mehrstellige Multiplikationen liefert. Höchst universelle Generierprozesse ermitteln aus einer definierten Menge an Zufallsmultiplikationen eine entsprechende Rechnung aus. Die zwei Faktoren einer Multiplikationsaufgabe werden immer per Zufall erzeugt. Dies führt zu einer nicht vorhersagbaren Aufgabengenerierung und erschwert die Manipulation von Lösungsfindungen. Zur Auswertung von Beantwortungen von Multiplikationen werden alle Benutzereingaben mitgespeichert, analysiert und umfangreiche Fehlerzustände ermittelt. Diese sind für Lehrkräfte auf einen Blick erkennbar und ermöglichen dadurch schnelle, korrekte und individuelle Lerneinschnitte für Schüler mit auftretenden Lernschwierigkeiten. Die spezielle Einfärbung von Fehlerzuständen in der Statistikauswertung ist dahingehend sehr hilfreich. Das Layout der Programmoberfläche ist volks- und mittelschulgerecht gestaltet, wobei das Design in absehbarer Zeit eine professionelle Überholung von Experten erfährt.

Verbesserungen des Mathe-Multi-Trainers sind in der Beispielgebung von Aufgabenstellungen mit dreistelligen Multiplikatoren angesiedelt. Erwähnte Faktoren beinhalten im gegenwärtigen Zustand für jede Aufgabenstellung dieser Kompetenzkategorie die Ziffer 1. Obwohl derartige Multiplikationen bereits sehr komplex sind, stellt der Wert 1 trotzdem eine gewisse Beeinträchtigung bei der Generierung dieser Problemstellungen dar. Da eine wiederholte Ausführung einer Ziffer im Faktor denselben Effekt hätte und Werte größer als 3 bereits zu größeren Einschränkungen des entsprechenden Multiplikanden führen würden, konnte dahingehend keine zufriedenstellende Lösung gefunden werden. Ein weiterer möglicher Änderungsvorschlag umfasst die Behandlung der Ergebnisrückgabe in der Benutzeroberfläche. Zur Zeit werden für Schüler positive Beantwortungen des Gesamtergebnisses auch bei zuvor aufgetretenen Rechenfehlern als korrekt gekennzeichnet. Diese Vorgehensweise ist zwar recht unwahrscheinlich und wurde deshalb im Prototyp vernachlässigt, sie könnte aber dennoch zu etwaigen Missverständnissen führen.

Die Applikation wurde bis zum jetzigen Zeitpunkt nur von ausgewählten Benutzern verwendet. Gleichwohl bleibt das Ziel bestehen, den Mathe-Multi-Trainer in den realen Volks- beziehungsweise Mittelschulunterricht einzubinden. Schüler und Lehrkräfte werden eine zuverlässige Lernanwendung vorfinden, die bei der Erlernung von mehrstelligen Multiplikationen unterstützend zur Seite steht.

## Literaturverzeichnis

- [Bader-Natal und Lotze 2011] BADER-NATAL, A.; LOTZE, T.: Evolving a learning analytics platform. In: *LAK '11 Proceedings of the 1st International Conference on Learning Analytics and Knowledge*. The Technology Enhanced Knowledge Research Institute, Athabasca University, Canada (Veranst.), ACM, 2011, S. 180–185 III, 3, 8, 9
- [Bakharia und Dawson 2011] BAKHARIA, A.; DAWSON, S.: SNAPP: A Bird's-Eye View of Temporal Participant Interaction. In: *LAK '11 Proceedings of the 1st International Conference on Learning Analytics and Knowledge*. The Technology Enhanced Knowledge Research Institute, Athabasca University, Canada (Veranst.), ACM, 2011, S. 168–173 6, 17, 18
- [Blikstein 2011] BLIKSTEIN, P.: Using learning analytics to assess students' behavior in open-ended programming tasks. In: *LAK '11 Proceedings of the 1st International Conference on Learning Analytics and Knowledge*. The Technology Enhanced Knowledge Research Institute, Athabasca University, Canada (Veranst.), ACM, 2011, S. 110–116 III, 20, 22, 23
- [Brooks et al. 2011] BROOKS, C.; EPP, C. D.; LOGAN, G.; GREER, J.: The Who, What, When, and Why of Lecture Capture. In: *LAK '11 Proceedings of the 1st International Conference on Learning Analytics and Knowledge*. The Technology Enhanced Knowledge Research Institute, Athabasca University, Canada (Veranst.), ACM, 2011, S. 86–92 23, 24
- [Clow und Makriyannis 2011] CLOW, D.; MAKRIYANNIS, E.: iSpot Analysed: Participatory Learning and Reputation. In: *LAK '11 Proceedings of the 1st International Conference on Learning Analytics and Knowledge*. The Technology Enhanced Knowledge Research Institute, Athabasca University, Canada (Veranst.), ACM, 2011, S. 34–43 16, 17
- [De Liddo et al. 2011] DE LIDDO, A.; SHUM, S. B.; QUINTO, I.; BACHLER, M.; CANNAVACCIUOLO, L.: Discourse-Centric Learning Analytics. In: *LAK '11 Proceedings of the 1st International Conference on Learning Analytics and Knowledge*. The Technology Enhanced Knowledge Research Institute, Athabasca University, Canada (Veranst.), ACM, 2011, S. 23–33 III, 12, 13, 14, 18
- [Education and Culture DG 2010] EDUCATION AND CULTURE DG: *eTwinning - Allgemeiner Leitfaden*. Jänner 2010. – URL: [http://desktop.etwinning.net/library/uploads/resources/4882/etwinning\\_general\\_guidelines\\_2010\\_de.pdf](http://desktop.etwinning.net/library/uploads/resources/4882/etwinning_general_guidelines_2010_de.pdf). – [Zugriff am 20.05.2012] 19
- [Eigenwillig und Mehlhorn 2008] EIGENWILLIG, A.; MEHLHORN, K.: Multiplikation langer Zahlen (schneller als in der Schule). In: *Taschenbuch der Algorithmen*. Springer, 2008 (eXamen.press), Kap. 11, S. 109–118 25
- [Fancsali 2011] FANCSALI, S. E.: Variable Construction for Predictive and Causal Modeling of Online Education Data. In: *LAK '11 Proceedings of the 1st International Conference on Learning*

- Analytics and Knowledge*. The Technology Enhanced Knowledge Research Institute, Athabasca University, Canada (Veranst.), ACM, 2011, S. 54–63 6, 12
- [Ferguson und Shum 2011] FERGUSON, R.; SHUM, S. B.: Learning Analytics To Identify Exploratory Dialogue within Synchronous Text Chat. In: *LAK '11 Proceedings of the 1st International Conference on Learning Analytics and Knowledge*. The Technology Enhanced Knowledge Research Institute, Athabasca University, Canada (Veranst.), ACM, 2011, S. 99–103 14
- [jQuery Foundation 2012] FOUNDATION jQuery: *jQuery: The Write Less, Do More, JavaScript Library*. 2012. – URL: <http://jquery.com/>. – [Zugriff am 22.07.2012] 41
- [Fournier et al. 2011] FOURNIER, H.; KOP, R.; SITLIA, H.: The Value of Learning Analytics to Networked Learning on a Personal Learning Environment. In: *LAK '11 Proceedings of the 1st International Conference on Learning Analytics and Knowledge*. The Technology Enhanced Knowledge Research Institute, Athabasca University, Canada (Veranst.), ACM, 2011, S. 104–109 3, 6, 9, 17
- [Gerber et al. 1994] GERBER, M. M.; SEMMEL, D. S.; SEMMEL, M. I.: Computer-Based Dynamic Assessment of Multidigit Multiplication. In: *Exceptional Children* 61 (1994), S. 114–125 V, 26, 27, 29
- [Graf et al. 2011] GRAF, S.; IVES, C.; RAHMAN, N.; FERRI, A.: AAT - A Tool for Accessing and Analysing Students' Behaviour Data in Learning Systems. In: *LAK '11 Proceedings of the 1st International Conference on Learning Analytics and Knowledge*. The Technology Enhanced Knowledge Research Institute, Athabasca University, Canada (Veranst.), ACM, 2011, S. 174–179 15
- [Grau-Valldosera und Minguillón 2011] GRAU-VALLDOSERA, J.; MINGUILLÓN, J.: Redefining dropping out in online higher education: a case study from the UOC. In: *LAK '11 Proceedings of the 1st International Conference on Learning Analytics and Knowledge*. The Technology Enhanced Knowledge Research Institute, Athabasca University, Canada (Veranst.), ACM, 2011, S. 75–80 15, 16
- [Lockyer und Dawson 2011] LOCKYER, L.; DAWSON, S.: Learning Designs and Learning Analytics. In: *LAK '11 Proceedings of the 1st International Conference on Learning Analytics and Knowledge*. The Technology Enhanced Knowledge Research Institute, Athabasca University, Canada (Veranst.), ACM, 2011, S. 153–156 3, 4, 17, 18
- [Merhbi 2011] MERHBI, A.: *Visualizing an LMs Discussion Forum: an insight into student/teacher behavior*. Juli 2011. – URL: <http://call4teachers.blogspot.com/2011/07/visualizing-lms-discussion-forum.html>. – [Zugriff am 28.05.2012] III, 17, 18
- [Miller und Bentley 2002] MILLER, R.; BENTLEY, T. ; SCHOOL LEADERSHIP, National C. for (Hrsg.): *'Unique Creation' - Possible futures - Four scenarios for 21st century schooling*. National College for School Leadership, 2002 4
- [Niemann et al. 2011] NIEMANN, K.; SCHMITZ, H.-C.; SCHEFFEL, M.; WOLPERS, M.: Usage Contexts for Object Similarity: Exploratory Investigations. In: *LAK '11 Proceedings of the 1st International Conference on Learning Analytics and Knowledge*. The Technology Enhanced Knowledge Research Institute, Athabasca University, Canada (Veranst.), ACM, 2011, S. 81–85 6, 9



- [Oracle 2012] ORACLE: *What is MySQL?* 2012. – URL: <http://dev.mysql.com/doc/refman/5.0/en/what-is-mysql.html>. – [Zugriff am 21.07.2012] 40
- [Pardo und Kloos 2011] PARDO, A.; KLOOS, C. D.: Stepping out of the box. Towards analytics outside the Learning Management System. In: *LAK '11 Proceedings of the 1st International Conference on Learning Analytics and Knowledge*. The Technology Enhanced Knowledge Research Institute, Athabasca University, Canada (Veranst.), ACM, 2011, S. 163–167 3, 8
- [Pham et al. 2012] PHAM, M. C.; CAO, Y.; PETRUSHYNA, Z.; KLAMMA, R.: Learning Analytics in a Teachers' Social Network. In: *Proceedings of the 8th International Conference on Networked Learning*, 2012, S. 414–421 19
- [Richards und DeVries 2011] RICHARDS, G.; DEVRIES, I.: Revisiting Formative Evaluation: Dynamic Monitoring for the Improvement of Learning Activity Design and Delivery. In: *LAK '11 Proceedings of the 1st International Conference on Learning Analytics and Knowledge*. The Technology Enhanced Knowledge Research Institute, Athabasca University, Canada (Veranst.), ACM, 2011, S. 157–162 15
- [Romero und Ventura 2010] ROMERO, C.; VENTURA, S.: Educational Data Mining: A Review of the State of the Art. In: *IEEE Transactions on Systems, Man, And Cybernetics, Part C: Applications and Reviews* 40 (2010), S. 601–618 20, 21
- [Rosen et al. 2011] ROSEN, D.; MIAGKIKH, V.; SUTHERS, D.: Social and Semantic Network Analysis of Chat Logs. In: *LAK '11 Proceedings of the 1st International Conference on Learning Analytics and Knowledge*. The Technology Enhanced Knowledge Research Institute, Athabasca University, Canada (Veranst.), ACM, 2011, S. 134–139 22
- [Scandura 2011] SCANDURA, J. M.: What TutorIT Can Do Better Than a Human and Why: Now and in the Future. In: *Technology, Instruction, Cognition and Learning (TICL)* 8 (2011), S. 175–227 III, 31, 32, 34, 35, 37
- [Schütze und Graën 2006] SCHÜTZE, H.; GRAËN, J.: *Die Programmiersprache PHP: Grundlagen und Anwendungen*. Mai 2006. – URL: <http://www.ims.uni-stuttgart.de/lehre/teaching/2006-SS/php/grundlagen.pdf>. – [Zugriff am 21.07.2012] 40
- [Sharkey 2011] SHARKEY, M.: Academic Analytics Landscape at the University of Phoenix. In: *LAK '11 Proceedings of the 1st International Conference on Learning Analytics and Knowledge*. The Technology Enhanced Knowledge Research Institute, Athabasca University, Canada (Veranst.), ACM, 2011, S. 122–126 III, 4
- [Shum und Ferguson 2011] SHUM, S. B.; FERGUSON, R.: *Social Learning Analytics / Knowledge Media Institute & Institute of Educational Technology, The Open University*. 2011. – Forschungsbericht III, 4, 5, 24
- [Siemens 2010a] SIEMENS, G.: *Learning Analytics & Knowledge*. Juli 2010. – URL: <https://tekri.athabascau.ca/analytics/>. – [Zugriff am 26.10.2011] 3
- [Siemens 2010b] SIEMENS, G.: *What are Learning Analytics?* August 2010. – URL: <http://www.elearnspace.org/blog/2010/08/25/what-are-learning-analytics/>. – [Zugriff am 26.10.2011] 3

- [Suthers und Rosen 2011] SUTHERS, D.; ROSEN, D.: A Unified Framework for Multi-Level Analysis of Distributed Learning. In: *LAK '11 Proceedings of the 1st International Conference on Learning Analytics and Knowledge*. The Technology Enhanced Knowledge Research Institute, Athabasca University, Canada (Veranst.), ACM, 2011, S. 64–74 V, 21, 22
- [Taraghi und Ebner 2012] TARAGHI, B.; EBNER, M.: Personal Learning Environment. In: *K. Wilbers & A. Hohenstein (Hrsg.), Handbuch E-Learning. Expertenwissen aus Wissenschaft und Praxis - Strategien, Instrumente, Fallstudien*. Wolters Kluwer Deutschland, 2012, S. 1–4 6
- [Teplovs et al. 2011] TEPLOVS, C.; FUJITA, N.; VATRAPU, R.: Generating Predictive Models of Learner Community Dynamics. In: *LAK '11 Proceedings of the 1st International Conference on Learning Analytics and Knowledge*. The Technology Enhanced Knowledge Research Institute, Athabasca University, Canada (Veranst.), ACM, 2011, S. 147–152 17, 20
- [Vatrapu 2011] VATRAPU, R.: Cultural Considerations in Learning Analytics. In: *LAK '11 Proceedings of the 1st International Conference on Learning Analytics and Knowledge*. The Technology Enhanced Knowledge Research Institute, Athabasca University, Canada (Veranst.), ACM, 2011, S. 127–133 14, 19
- [Vatrapu et al. 2011] VATRAPU, R.; TEPLOVS, C.; FUJITA, N.; BULL, S.: Towards Visual Analytics for Teachers' Dynamic Diagnostic Pedagogical Decision-Making. In: *LAK '11 Proceedings of the 1st International Conference on Learning Analytics and Knowledge*. The Technology Enhanced Knowledge Research Institute, Athabasca University, Canada (Veranst.), ACM, 2011, S. 93–98 III, 6, 7
- [Verbert et al. 2011] VERBERT, K.; DRACHSLER, H.; MANOUSELIS, N.; WOLPERS, M.; VUORIKARI, R.; DUVAL, E.: Dataset-driven Research for Improving Recommender Systems for Learning. In: *LAK '11 Proceedings of the 1st International Conference on Learning Analytics and Knowledge*. The Technology Enhanced Knowledge Research Institute, Athabasca University, Canada (Veranst.), ACM, 2011, S. 44–53 10, 11
- [Weyl 2008] WEYL, Estelle: *jQuery Tutorial: DOM Manipulation and sorting*. November 2008. – URL: <http://www.evotech.net/blog/2008/11/jquery-tutorial-dom-manipulation-and-sorting/>. – [Zugriff am 22.07.2012] 41
- [Woodward und Howard 1994] WOODWARD, J.; HOWARD, L.: The Misconceptions of Youth: Errors and Their Mathematical Meaning. In: *Exceptional Children* 61 (1994), S. 126–136 29
- [Zend 2012a] ZEND: *Programmer's Reference Guide - Übersicht*. 2012. – URL: <http://framework.zend.com/manual/de/introduction.overview.html>. – [Zugriff am 21.07.2012] 41
- [Zend 2012b] ZEND: *Zend Framework & MVC Introduction*. 2012. – URL: <http://framework.zend.com/manual/en/learning.quickstart.intro.html>. – [Zugriff am 21.07.2012] 41