



Das E-Assessment-System EASy: Computerunterstützter Übungsbetrieb im Informatikstudium

Autorin und
Autoren: Susanne Gruttmann,
Claus Usener,
Herbert Kuchen
Portalbereich: Aus der Praxis
Stand: 01.02.2010

Inhaltsverzeichnis

1. Einleitung.....	1
2. Das E-Assessment System EASy	2
2.1. Multiple-Choice-Aufgaben mit EASy	4
2.2. Das EASy-Modul für mathematische Beweise	5
2.3. Verifikationsbeweise mit EASy.....	7
2.4. Programmieraufgaben mit EASy.....	9
3. Vorlesungsbegleitende Übungen mit EASy	11
3.1. Assessment-Prozesse mit EASy	11
3.2. Evaluation des Einsatzes von EASy	13
4. Fazit	15
Literaturverzeichnis	16
Die Autorin / die Autoren	17

1. Einleitung

Das Informatikstudium soll analytische, kreative und konstruktive Fähigkeiten zur Neu- und Weiterentwicklung von Soft- und Hardware-Systemen vermitteln. Die Studierenden sollen zudem dazu befähigt werden, grundlagen- und anwendungsorientierte Forschung auf dem Gebiet der Informatik durchzuführen (GI, 2005). Um diese Fähigkeiten und Fertigkeiten zu erlernen, sind im Informatikstudium neben einfachen kognitiven Lehr- und Lernzielen, wie z. B. der Vermittlung von deklarativem Wissen in Form von Faktenwissen, vor allem Lehr- und Lernziele höherer kognitiver Ordnung von Bedeutung. Die meisten informatischen Fähigkeiten lassen sich nur durch sorgfältig betreutes Üben erwerben (ASIIN, 2006; GI, 2005). Das theoretisch erlernte Wissen kann durch die Bearbeitung von geeigneten Aufgaben reflektiert und verinnerlicht werden (Dyckhoff et al., 2008). Daher ist der regelmäßige Übungsbetrieb als eine zentrale Aufgabe innerhalb der Lehr-, Lern- und Prüfungsprozesse des Informatikstudiums an Hochschulen zu werten. Hohe Studierendenzahlen und die zunehmende Knappheit von finanziellen Ressourcen erschweren jedoch die Durchführung regelmäßiger Übungen. Ein vielversprechendes Instrument, den Aufwand zu reduzieren und dennoch langfristig einen hochwertigen Übungsbetrieb zu gewährleisten, ist im Einsatz von E Assessment-Systemen zu sehen.

In der Informatik zählen neben Wissensabfragen insbesondere Programmieraufgaben, Modellierungsaufgaben, Berechnungen und mathematische Beweise zu den relevanten Aufgabenarten. Von den aktuell am Markt etablierten Systemen werden jedoch meistens nur einfache elektronische Aufgabentypen wie Multiple-Choice-Aufgaben, Zuordnungsaufgaben oder Lückentexte unterstützt. Diese eignen sich zwar zur Über-



prüfung von Wissenszuwachsen, kognitive Fähigkeiten und Methodenwissen können jedoch nur bedingt durch diese Aufgabentypen abgeprüft werden (Chalmers & McAusland, 2002). Komplexere Aufgaben mit diesen Aufgabenarten, die vom Lernenden zunächst die Analyse und Synthese von Wissen fordern, fragen im Regelfall letztlich meist nur das Endergebnis ab, z. B. das numerische Ergebnis einer Berechnung. Der Lösungsweg, also wie das Endergebnis entstanden ist, wird jedoch nicht evaluiert. Für das fachgerechte Überprüfen komplizierter Zusammenhänge und praktischer Fähigkeiten in der Informatik eignen sich diese simplen Aufgabentypen daher nur bedingt. Die Kontrolle und Bewertung von Aufgaben, bei denen die Anwendung analytischer, kreativer und konstruktiver Fähigkeiten gefordert ist, wird von der Mehrzahl der heute am Markt verbreiteten Systeme nicht unterstützt (Gruttmann et al., 2008b). Um den curricularen Zielen des Informatikstudiums Rechnung zu tragen und dennoch vom Einsatz elektronischer Systeme zur Überprüfung von Lernfortschritten profitieren zu können, wurde an der WWU Münster das E Assessment-System EASy entwickelt, das im Rahmen dieses Aufsatzes thematisiert wird. EASy (als Akronym für E Assessment-System) soll den Anforderungen des Lehrgebiets entsprechend einen grundlegenden Paradigmenwechsel von traditionellen papierbasierten zu computerunterstützten Übungen bewirken. Durch die Bereitstellung einfacher Aufgabentypen wie Multiple-Choice, aber auch anspruchsvoller Aufgabentypen zu mathematischen Beweisen oder Programmierung soll die Durchführung der Übungen in hoher Qualität sichergestellt werden.

Im vorliegenden Beitrag werden anhand des E Assessment-Systems EASy die Potenziale einer Computerunterstützung des Übungsbetriebs im Informatikstudium thematisiert. Hierzu werden in Kapitel 2 zunächst die Gestaltung, Funktionalität und Handhabung des Systems sowie der verschiedenen im System integrierten Aufgabenmodule vorgestellt. Aktuell bietet das System Module für Multiple-Choice-Aufgaben, Programmieraufgaben, mathematische Beweise und Verifikationsbeweise an. Anhand des EASy-Aufgabenmoduls für mathematische Beweise wird in Kapitel 3 der Praxiseinsatz des Systems demonstriert, wodurch die resultierenden Potenziale des Einsatzes digitaler Medien in den Prozessen des Übungsbetriebs verdeutlicht werden. Die Ergebnisse einer empirischen Evaluation des Einsatzes von EASy unterstreichen die Relevanz des Ansatzes und erlauben Rückschlüsse auf generelle Optimierungspotenziale durch das E Assessment. Der Beitrag schließt mit einer Zusammenfassung und einem Ausblick auf weitere Entwicklungen.

2. Das E-Assessment System EASy

Durch die Entwicklung von EASy wird angestrebt, ein einheitliches System für die Unterstützung des Übungsbetriebs der wesentlichen Informatik-Grundlagenveranstaltungen zu etablieren. In der aktuell vorliegenden Version ermöglicht die EASy-Webapplikation eine komfortable computerunterstützte Vorbereitung, Durchführung und Nachbereitung von Lernfortschrittskontrollen im Informatikstudium, indem es die typischen Prozesse des Übungsbetriebs an sinnvollen Stellen elekt-



ronisch unterstützt. Neben obligatorischen Verwaltungsfunktionen bietet das System Unterstützungsfunktionen für die Erstellung, Durchführung und Überprüfung verschiedener relevanter Aufgabenarten des Informatikstudiums. Hierfür wurden die Prozesse des traditionellen Übungsbetriebs unter Beachtung allgemeiner sowie aufgabentyp-spezifischer didaktischer, methodischer und organisatorischer Aspekte in eine computerunterstützte Form übertragen (Gruttmann & Kuchen, 2008).

Konzeptionell basiert das E Assessment-System EASy auf einer dreischichtigen, modularen und erweiterbaren Web-Plattform, die mit verhältnismäßig geringem Aufwand um Funktionsmodule für Aufgabentypen ergänzt werden kann (vgl. Abb. 1).

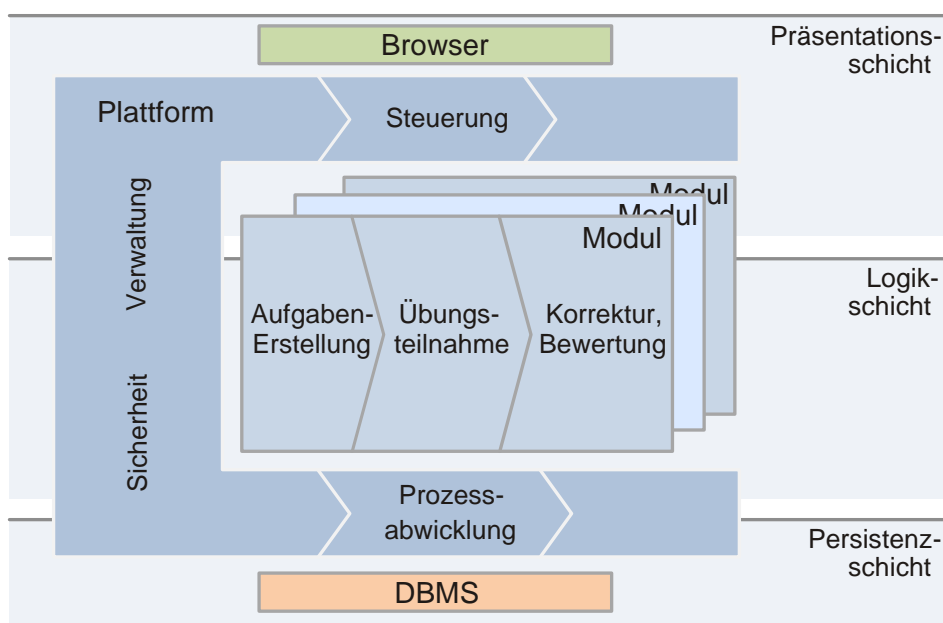


Abb. 1: Architektur der E-Assessment-Plattform EASy.

Im Architekturentwurf umrahmt die Plattform die aufgabenspezifischen Module auf Ebene der Präsentationsschicht und der Logikschicht. Die fachlich angemessene Ausgestaltung, wie eine Aufgabe präsentiert wird, wie sie vom Übungsteilnehmer zu bearbeiten ist und wie eine automatische Vorkorrektur erfolgt, fällt dabei in die Zuständigkeit des entsprechenden Aufgabenmoduls und ist unabhängig von der Implementierung der Web-Plattform. Dadurch wird sichergestellt, dass sich ein neues Aufgabenmodul an den fachspezifischen Charakteristiken des Aufgabentyps orientieren kann, ohne sich technischen Einschränkungen unterwerfen zu müssen.

Zur technischen Realisierung des Systems wird auf das Webframework JBoss Seam in Verbindung mit der JBoss Portlet Bridge zurückgegriffen, um die einzelnen Module und Funktionen des Systems als unabhängige Anwendungen in einem Portal betreiben zu können. Durch diese Kombination steht für die Implementierung der einzelnen Module eine Vielzahl an Technologien wie z. B. JavaServerFaces bzw. RichFaces und



AJAX bereit.

Derzeit existieren in EASy Aufgabenmodule für die Java-Programmieraufgaben, für mathematische Beweise bzw. Verifikationsbeweise mit der Hoare-Logik und für Multiple-Choice-Aufgaben. Eine Erweiterung der Plattform um zusätzliche Aufgabenmodule, z. B. für Diagramm- und Modellierungsaufgaben, einfache Berechnungen und Freitextaufgaben befindet sich in der Planung.

2.1. Multiple-Choice-Aufgaben mit EASy

Multiple-Choice-Aufgaben können auch im Bereich informatischer Inhalte Anwendung finden. Zur Abrundung des Aufgabentyp-Angebots wurde ein entsprechendes Modul für die EASy-Plattform entwickelt. In Abbildung 2 wird ein Beispiel für eine Multiple-Choice-Aufgabe mit Mehrfachauswahl zur einfachen Wissensabfrage gezeigt, wie sie etwa in der Vorlesung Informatik I: Programmierung abgefragt werden könnte.

The screenshot shows the EASy platform interface. At the top, there is a navigation bar with the Westfälische Wilhelms-Universität Münster logo on the left and the 'Praktische Informatik' logo with the EASy character on the right. Below the navigation bar, there is a user profile section showing 'Angemeldet als: prinzPoldi' and a 'Logout' button. The main content area displays a question: 'Welche dieser Programmiersprachen sind dem Paradigma der deklarativen Programmierung zuzuordnen?'. The question is followed by a list of programming languages with checkboxes: BASIC (checked), Cobol (unchecked), Curry (checked), Fortran (unchecked), Haskell (checked), Perl (unchecked), Prolog (checked), and Scheme (checked). On the left side of the question, there is an 'info' box containing the title 'MC - Paradigmen Programmiersprachen' and the score 'erreichbare Punkte 7.5'. At the top of the question area, there are 'zurück' and 'speichern' buttons.

Abb. 2: Einfache Wissensabfrage mit Multiple-Choice

In diesem Fall müssen die Prüfungsteilnehmer aus acht Antwortoptionen die korrekte Lösungsmenge bestimmen. Konkret werden sie angehalten, die Zugehörigkeit verschiedener Programmiersprachen zu einem bestimmten Paradigma anzugeben. Sie müssen dafür die verschiedenen Antwortalternativen bzw. deren Charakteristik kennen. Mit Aufgaben dieser Art wird geprüft, inwiefern sich die Teilnehmer gewisse Definitionen oder Sachverhalte angeeignet haben oder sich diese merken können.



Obwohl das Ausmaß an Aktivität und Kreativität der Studierenden bei der Bearbeitung von Aufgaben dieses Typs verhältnismäßig gering ist, kann der Einsatz für bestimmte Zwecke in Veranstaltungen des Informatikstudiums zielführend sein. Da im Informatikstudium neben der Vermittlung von Faktenwissen vor allem die Entwicklung Fähigkeiten und Fertigkeiten im Vordergrund steht, ist eine ausschließliche Anwendung von Multiple-Choice-Aufgaben in Lernfortschrittskontrollen jedoch ungeeignet. Aus diesem Grund wurden neben dem Aufgabenmodul für Multiple-Choice weitere Module realisiert, die die Bearbeitung und Überprüfung analytischer, kreativer und konstruktiver Aufgaben des Informatikstudiums ermöglichen.

2.2. Das EASy-Modul für mathematische Beweise

Vertieftes mathematisches Verständnis, das in Hochschulen vermittelt werden soll, zeigt sich insbesondere in der Fähigkeit, mathematische Zusammenhänge selbständig erschließen und konstruktiv beweisen zu können (GI, 2005). Mathematische Beweise stellen daher einen klassischen Aufgabentyp in Lernfortschrittskontrollen vieler mathematisch-technischer Studiengänge dar. Im Fach Informatik werden sie z. B. in Vorlesungen wie Datenstrukturen und Algorithmen für Aufgaben zur Gültigkeit einer Laufzeitabschätzung benutzt. Die derzeit verfügbaren E-Assessment-Systeme stellen jedoch keine adäquate Unterstützung für Aufgaben zu mathematischen Beweisen bereit.

An ein E-Assessment-System zur Erstellung, Bearbeitung und Korrektur von mathematischen Beweisaufgaben kann eine Reihe fachlicher Anforderungen formuliert werden (Gruttmann et al., 2008b). So sollte ein entsprechendes System das schrittweise Umformen von Termen durch Anwendung von Regeln ermöglichen und die mathematische Korrektheit der Regelanwendungen überwachen.

Es muss ferner die relevanten Beweisstrategien des Anwendungskontexts bereitstellen (Induktionen, Fallunterscheidungen etc.) und im Bedarfsfall um weitere Strategien, Regeln und vorauszusetzende Sätze erweiterbar sein.

Das EASy-Modul für mathematische Beweise wurde als eigenständiges, aufgabenindividuelles Modul entsprechend dieser fachlichen Anforderungen konzipiert. Die komplexe Funktionalität des EASy-Beweismoduls wurde mit Hilfe eines Java-Applets zugänglich gemacht, das nahtlos in die EASy-Plattform eingebettet ist. Damit das Applet die Infrastrukturdienste der EASy-Plattform nutzen kann, stellt die Plattform entsprechende Schnittstellen für die Frontend- und Backend-Komponenten des Applets bereit. Im Kopf der EASy-Benutzeroberfläche werden die zentralen Infrastrukturdienste der Plattform bereitgestellt, im Rumpf wird das Applet für die Präsentation, Bearbeitung und Begutachtung von Beweisen geladen (siehe Abb. 3).



WESTFÄLISCHE WILHELMS-UNIVERSITÄT MÜNSTER

Praktische Informatik

EASy

Logout

Angemeldet als: prinzPoldi

→ Allgemein → Student

→ Veranstaltungen → Aufgaben → Ergebnisse

zurück

Titel: Musteraufgabe 1 erreichbare Punkte: 30,0

Hinweis: Klicken Sie zum Speichern auf 'Beweis absenden'. Sie werden zur Eingabe Ihrer Anmeldeinformationen aufgefordert.

Beweisaktionen

Aktionen

- Beweis absenden
- Beweis drucken
- Beweis zurücksetzen
- Letzte Strategie löschen
- Schritt zurück

Ausführen

Strategie hinzufügen

Induktion

- Boolesche Umformung
- Arithmetische Umformung
- Es-Existiert Aussage definieren
- Vorbedingungen => Konklusion
- Konklusion => Vorbedingungen
- Fallunterscheidung
- Aufspaltung der Folgerung
- Abschätzungskette
- Verifikationsbeweis (Hoare-Logik)

Hinzufügen

Beweisübersicht

- aufgabe_1
 - Induktion nach n
 - Induktionsanfang: aufgabe_1
 - Boolesche Umformung

Abb. 3: Benutzeroberfläche des Moduls für mathematische Beweise

Gestaltung und Handhabung: Die Benutzeroberfläche des Applets ist in mehrere Bereiche aufgeteilt. In der Mitte werden das Theorem, das bewiesen werden soll, sowie der gegenwärtige Status der Beweisführung angezeigt. Auf der linken Seite der Benutzeroberfläche befinden sich elementare administrative Funktionen zum Speichern, Drucken oder Zurücksetzen von Beweisen. Zudem kann in diesem Bereich die Auswahl von Beweisstrategien erfolgen. Mögliche Strategien sind z. B. Induktionen, Fallunterscheidungen oder Abschätzungen. Nachdem der Benutzer sich für eine Beweisstrategie entschieden hat, erscheinen auf der rechten Seite Regeln zur Termumformung, mittels derer der Benutzer das Theorem dahingehend manipulieren kann, dass die Terme auf der linken und der rechten Seite des Gleichheitszeichens syntak-



tisch äquivalent sind. Die Auswahl des relevanten Teilterms, auf den eine Regel angewendet werden soll, erfolgt komfortabel mit Hilfe eines Termauswahlbaumes. Zur besseren Orientierung kann eine Beschreibung der gewählten Regel eingesehen werden.

Funktionsweise: Das Backend des Applets stellt den mathematischen Kern bereit, der auf Basis von Termersetzungen operiert (Baader & Nipkow, 1999). Jedes Theorem ist mit einer Reihe von Termersetzungsregeln ausgestattet, wobei jede dieser Regeln aus einer Prämisse und einer Konklusion besteht. Der mathematische Kern und seine integrierte Rule Engine stellen sicher, dass der aktuelle Beweiskontext den Prämissen einer Regel genügt. Ist dies nicht der Fall, wird die Anwendung der Regel verwehrt. Von den Studierenden selbst formulierte Umformungssätze sind separat mit Hilfe der verfügbaren Regeln zu beweisen, bevor sie von EASy als korrekt und benutzbar angesehen werden. Dieser Mechanismus garantiert die mathematische Korrektheit von EASy, da Studierende keine fehlerhaften Beweisschritte durchführen können und somit keine falschen Beweise produzieren können. Es kann lediglich der Fall auftreten, dass Studierende bei der Beweisentwicklung „steckenbleiben“. Aufgrund der großen Anzahl an Regeln in der Rule Engine ist es nicht möglich, die Beweise ohne das nötige mathematische Wissen nach dem Trial-and-Error-Prinzip zu lösen. Auf die Integration eines Computeralgebra-Systems oder eines automatischen Theorembeweisers wurde bewusst verzichtet, um den Studierenden die Umformungs- und Berechnungsarbeit nicht unfreiwillig an entscheidenden Stellen zu anzunehmen.

Das EASy-Modul für Beweisaufgaben unterstützt folglich eine prozessbasierte Überprüfung und eine intelligente automatische Bewertung studentischer Leistungen, wodurch es ermöglicht wird, die Gesamtheit der Beweisschritte zu beurteilen statt lediglich ein Endergebnis zu bewerten.

2.3. Verifikationsbeweise mit EASy

Das EASy-Modul für Beweisaufgaben wurde ursprünglich für klassische mathematische Beweise wie z. B. zur vollständigen Induktion, Fallunterscheidungen, boolesche und arithmetische Umformungen oder Abschätzungsketten entwickelt. Durch eine funktionale Erweiterung wurde jüngst ein äquivalent operierendes Aufgabenmodul für Verifikationsbeweise mit der Hoare-Logik entwickelt. Es handelt sich hierbei um eine Methode zur Verifikation von Software, bei der durch eine Menge logischer Beweisregeln Aussagen über das Verhalten und die Korrektheit von Programmen bewiesen werden (Hoare, 1969). Einem Programm c werden dabei eine Vorbedingung $\{P\}$ und eine Nachbedingung $\{Q\}$ zugeordnet (z. B. $\{x \geq 0\}$ while $(x > 0)$ do $x \leftarrow x-1$ $\{x = 0\}$). Die Korrektheit dieses so genannten Hoare-Tripels $\{P\} c \{Q\}$ kann mit Hilfe eines fest definierten Regelsatzes, dem Hoare-Kalkül, sowie ergänzenden arithmetischen und booleschen Umformungen schrittweise bewiesen werden.

Die Hoare-Logik bildet einen wichtigen Bestandteil des Studiums der theoretischen Informatik (etwa in der Vorlesung Formale Spezifikation), dessen Methoden nicht nur



theoretisch erlernt, sondern auch praktisch erprobt werden müssen.

Um diesen Aufgabentyp mit EASy sinnvoll unterstützen zu können, waren konkrete funktionale und fachliche Anforderungen zu erfüllen. So mussten etwa zusätzlich zu arithmetischen und booleschen Termen eine neue Termart zur Abbildung von Programmfragmenten sowie eine kombinierte Termart für das Hoare-Tripel initialisiert werden. Ferner waren die Regeln des Hoare-Kalküls und eine entsprechende neue Beweisstrategie für Verifikationen bereitzustellen. Um den Einarbeitungsaufwand zu verringern, unterscheidet sich das Modul für diesen Aufgabentyp in Bezug auf Gestaltung, Funktionsweise und Handhabung nur unwesentlich vom ursprünglichen EASy-Beweismodul (vgl. Abb. 4).

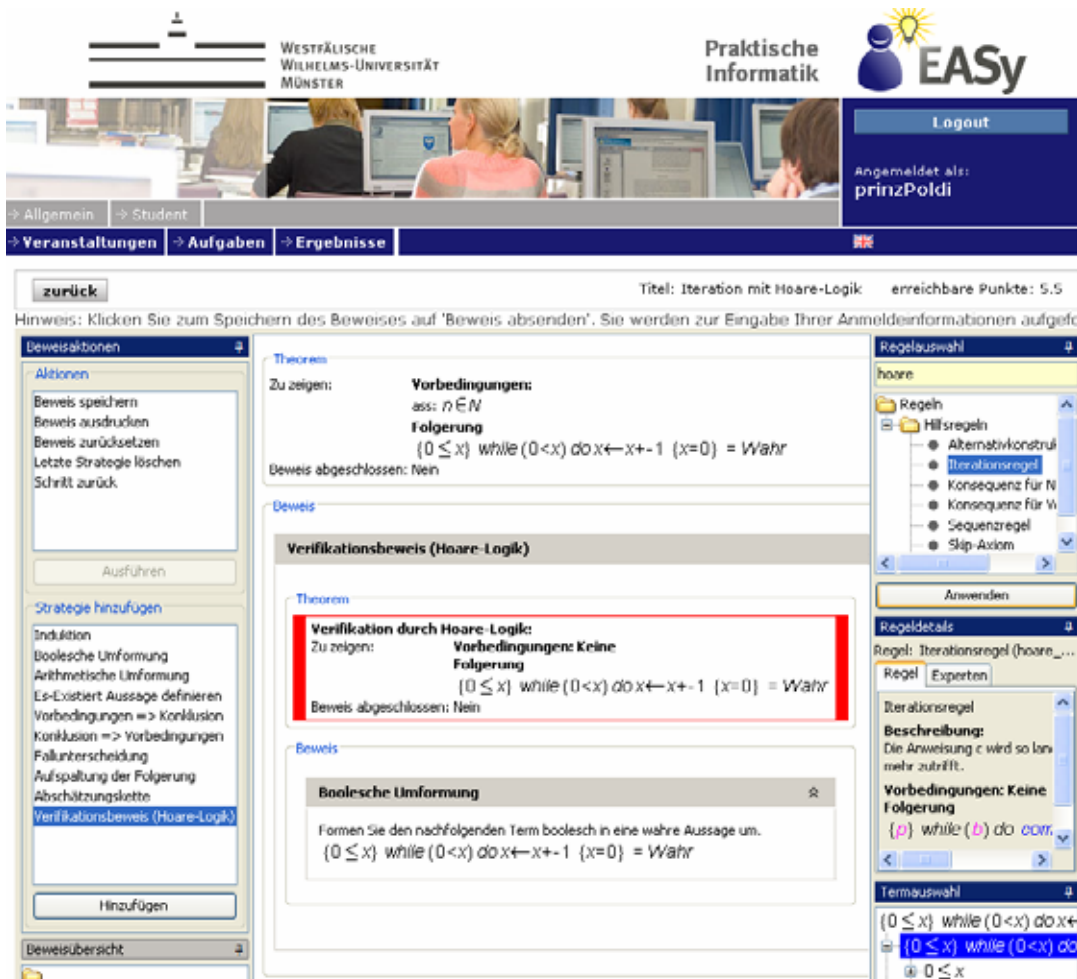


Abb. 4: Benutzeroberfläche des Moduls für Verifikationsbeweise

Dem Übungsteilnehmer wird das zu beweisende Hoare-Tripel in der Mitte der Benutzeroberfläche angezeigt. Nachdem er die Strategie „Verifikationsbeweis“ gewählt hat,



erscheinen im Regelauswahlbereich auf der rechten Seite des Applets neben diversen arithmetischen und booleschen Regeln nun auch die Regeln des Hoare-Kalküls, die dann auf ausgewählte Teilterme des Tripels angewendet werden können. Eine fehlerhafte Regelanwendung wird automatisch verhindert. Eine Systemnachricht informiert den Übungsteilnehmer in diesem Fall direkt über den Fehler. Insofern können, wie beim EASy-Modul für mathematische Beweise, auch mit dem Aufgabenmodul für Verifikationsbeweise nur korrekte Beweise erzeugt werden. Der Anspruch an den Übungsteilnehmer ist auch in diesem Modul, mit den vorhandenen Strategien und Regeln auf korrekte Weise einen Beweis zu führen. Wie bei traditionellen, auf Papier geführten Verifikationsbeweisen ist die entscheidende Schwierigkeit, das Theorem an geeigneten Stellen so zu manipulieren, dass die Anwendung der Regeln des Hoare-Kalküls zum gewünschten Ergebnis führt.

2.4. Programmieraufgaben mit EASy

In einer Grundlagenvorlesung zur Programmierung ist es essenziell, den Studierenden die Möglichkeit zu geben, das erlernte Wissen durch Programmieraufgaben praktisch anzuwenden und so die Programmierkonzepte und Grammatikregeln der Sprache zu internalisieren (Görlitz & Müller, 2002). Da eine vollständig manuelle Korrektur durch Lehrpersonal sehr viel Zeit in Anspruch nehmen kann – bei der manuellen Quellcodeinspektion wird dieser Zeitverbrauch mit 150 Codezeilen pro Stunde angegeben (Thaller, 2000) – ist eine technische Unterstützung bei der Korrektur zweckmäßig. Eine Computerunterstützung des Übungsbetriebs kann jedoch nicht nur bei der Korrektur hilfreich sein. Die elektronische Abgabe der Lösung eröffnet beispielsweise die Möglichkeit, vor der finalen Abgabe bereits bestimmte Basisüberprüfungen durchzuführen, die dem Studierenden die Gelegenheit bieten, den eigenen Lösungsansatz zu überdenken und Verbesserungen durchzuführen.

Das EASy-Modul für Programmieraufgaben ist zunächst für Aufgaben in der Programmiersprache Java entwickelt worden. Technisch greift das System bei der Kontrolle des Quellcodes auf dynamische und statische Analysen zurück. So werden vom Aufgabensteller vorgegebene JUnit-Testfälle zur Kontrolle der generellen Funktionalität der Abgabe, Stilanalysen zur Überprüfung der Einhaltung von gängiger Programmierkonventionen sowie eine Bytecode-Analyse zur Identifikation bekannter Fehlermuster durchgeführt. Schließlich können in einer Plagiarismuskontrolle Duplikate identifiziert werden.

Aufgabe erstellen: Grundlage für die Programmieraufgaben, die mit EASy bearbeitet werden sollen, ist eine textuelle Aufgabenstellung, in der der Dozent den Programmierauftrag und gewisse Rahmendaten (wie z. B. vorgegebene Klassennamen) bekannt gibt. Eine entsprechende Aufgabe kann in EASy einfach konfiguriert werden, indem der Dozent die Musterlösung und relevante JUnit-Tests zur Vorüberprüfung und finalen Bewertung der Einreichungen einstellt.



Lösung einreichen: Die Aufgabenbearbeitung wird vom Studierenden auf dem eigenen Computer durchgeführt, so dass der Umgang mit einer realen Entwicklungsumgebung (IDE) geschult wird. Gleichzeitig wird der Studierende nicht durch eine vom System vorgegebene Entwicklungsumgebung bevormundet und kann die von ihm bevorzugte IDE verwenden. Nachdem der Studierende das Ergebnis seiner Programmierarbeit über die Bearbeitungsansicht in das System geladen hat, kann er es mit Hilfe einer Vorabüberprüfung testen. Bei dieser Prüfung wird der eingereichte Quellcode auf Kompilierbarkeit überprüft und mittels der zuvor vom Aufgabenersteller definierten Testfälle einem grundlegenden Funktionstest unterzogen. Auftretende Fehler kann der Studierende direkt beheben und sich so langsam an eine akzeptable Lösung herantasten. Studierenden wird in EASy die Möglichkeit gegeben, eigene JUnit-Testfälle hochzuladen. Dadurch können sie zum einen ihre Kompetenzen in der Erstellung von Testfällen schulen, zum anderen wird dadurch sukzessive die Basis verfügbarer Testfälle erweitert und die Genauigkeiten der dynamischen Tests erhöht. Abbildung 5 zeigt eine typische Bearbeitungsansicht nach der Durchführung eines Vorabtests.

The screenshot shows the EASy web interface for submitting and testing code. At the top, there is a navigation bar with 'Veranstaltungen', 'Aufgaben', and 'Ergebnisse'. The user is logged in as 'prinzPoldi'. The main content area is divided into several sections:

- Hochgeladene Dateien:** A table listing uploaded files with columns for ID, name, type, and date.

ID	name	enthalten Typen	Datum	
2	Naturals.java [notwendige Datei]	[Naturalis]	20.01.10 13:52:49	Loeschen
5	Filter.java [notwendige Datei]	[Filter]	20.01.10 13:54:47	Loeschen
9	Evens.java [notwendige Datei]	[Evens]	20.01.10 13:59:38	Loeschen
8	EvaluationTest.java [JUnit Datei]	[EvaluationTest]	20.01.10 13:55:57	Loeschen
- Pruefen:** Radio buttons for 'Loesung' (selected) and 'JUnit-Test'.
- Testergebnis:** A detailed view of the test results, including:
 - Erstelldatum:** 20.01.10 13:56:03
 - Fehlende Dateien:** Es sind alle notwendigen Dateien hinzugefuegt worden.
 - Compiler-Fehler:** Es sind keine Fehler aufgetreten.
 - Fehlerhafte Struktur:**
 - File: Naturals.java: public class Naturals implements Sequence. Gefordertes Feld: int current;
 - File: Evens.java: public class Evens extends Filter. Geforderte Methode: void reset();
 - JUnit-Testergebnis:** A table showing the results of individual test cases.

Testsuite	Testcase	Status	Fehlermeldung
PreparatoryTest	NaturalsTestNext	success	
PreparatoryTest	NaturalsTestReset	success	
PreparatoryTest	NaturalsTest	success	

Abb. 2: Benutzeroberfläche zur Abgabe von Programmieraufgaben



Korrektur: Die Korrektur findet in mehreren Phasen statt: Zunächst werden alle Abgaben anhand der zuvor definierten statischen und dynamischen Analysen automatisch überprüft. In einem zweiten automatisierten Schritt werden alle Einreichungen einer Plagiarismuskontrolle unterzogen. Sofern die Aufgabenerstellung die Abgabe eigener Testfälle erlaubt, werden in dieser Phase ebenfalls die von den Studierenden eingereichten Testfälle auf die Abgaben der anderen Studierenden angewendet, um besonders gute Testfälle herauszufiltern. In der dritten Korrekturphase hat nun der Tutor Zugriff auf die Abgaben, um sie in einem letzten Schritt einer manuellen Kontrolle zu unterziehen. Eine manuelle Bewertung ist notwendig, da nicht alle Bewertungskriterien automatisch überprüft werden können (z. B. die sinnvolle Benennung von Variablen und Methoden). Unterstützung erhält der Tutor bei der manuellen Kontrolle durch die Ergebnisse der vorherigen Phasen. Die Ergebnisse der Korrekturphase kann der Studierende nach Abschluss der Korrekturen auf der Plattform einsehen.

Das EASy-Modul für Programmieraufgaben bietet gegenüber dem traditionellen Bearbeitungs- und Korrekturprozess den Vorteil, dass einerseits der Studierende anhand der Vorabüberprüfung schon bei der Abgabe der Aufgabe zum Überdenken seiner Lösung angehalten wird. Andererseits kann der Tutor durch die automatische Korrektur eingesparte Zeit zur intensiveren Betreuung der Studierenden verwenden. Gleichzeitig wird durch die automatische Korrektur eine bessere Vergleichbarkeit der Lösungen garantiert, da diese bei allen Abgaben identisch ausfällt.

3. Vorlesungsbegleitende Übungen mit EASy

Studien belegen, dass es sinnvoll ist, neue elektronische Lehr- und Lernformen äquivalent zu gängigen traditionellen Methoden zu gestalten, damit sie nachhaltig an den Universitäten etabliert werden können (Dyckhoff et al., 2008; Kleimann & Wannemacher, 2005). Daher orientieren sich die webbasierten Prozesse des Übungsbetriebs mit EASy an den traditionellen Abläufen.

3.1. Assessment-Prozesse mit EASy

EASy ermöglicht eine qualitativ hochwertige Durchführung von Lernfortschrittskontrollen im Informatikstudium auch für große Studierendenzahlen, indem es die typischen Prozesse des Übungsbetriebs an sinnvollen Stellen elektronisch unterstützt. Neben obligatorischen Verwaltungsfunktionen bietet das System Unterstützungsfunktionen für die Erstellung, Durchführung und Überprüfung von Übungsaufgaben. Hierfür wurden die Prozesse des traditionellen Übungsbetriebs unter Beachtung allgemeiner sowie aufgabentyp-spezifischer didaktischer, methodischer und organisatorischer Aspekte in eine computerunterstützte Form übertragen (vgl. Abb. 6).

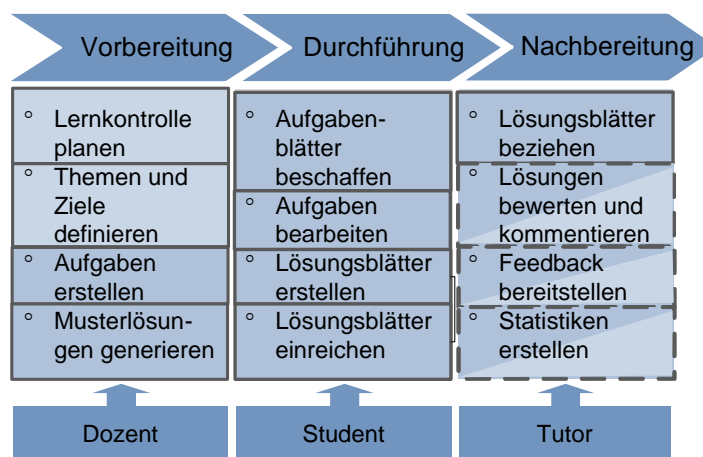


Abb. 3: Unterstützung der Prozesse im Übungsbetrieb durch EASy

Im Folgenden werden die Prozesse des Übungsbetriebs mit EASy am Beispiel einer mathematischen Beweisaufgabe verdeutlicht.

Vorbereitung: Mit Hilfe eines Aufgabeneditors können die Aufgaben komfortabel visuell, textuell bzw. in einer einfachen (mathematischen) Syntax erstellt und mit notwendigen Metadaten versehen werden. Jede Aufgabe beinhaltet neben dem zu beweisenden Theorem eine Menge an Standard-Regeln, kann zudem aber bei Bedarf mit weiteren aufgabenspezifischen Regeln ausgestattet werden. Indem der Lehrende den Beweis zu einer Aufgabe selbst führt und abspeichert, wird eine Musterlösung generiert. Einzelne Aufgaben können zu einem Übungsblatt zusammengefasst werden, mit Bewertungsvorgaben sowie zeitlichen und organisatorischen Rahmendaten für die Bearbeitung versehen werden. Das Übungsblatt wird schließlich den Studierenden einer ausgewählten Veranstaltung über das EASy-Gesamtsystem bereitgestellt.

Durchführung: Nachdem eine Übungsaufgabe ausgewählt wurde, können die Studierenden den geforderten Beweis innerhalb von EASy orts- und zeitunabhängig und sofern gestattet in Gruppenarbeit entwickeln. Das Applet stellt ihnen hierzu alle benötigten Regeln und Beweisstrategien bereit. Es ist den Studierenden gestattet, den Beweis beliebig oft zu wiederholen. Zwischenstände können gespeichert und wieder geladen werden. Als finales Ergebnis wird der bei Ablauf der Bearbeitungszeit zuletzt gespeicherte Stand des Beweises gewertet.

Nachbereitung: Nach Ablauf der Bearbeitungszeit erhalten die Tutoren automatisch die Lösungsversionen der ihnen zugeteilten Übungsteilnehmer. Sie können die Lösungen komfortabel in einem aufgabentyp-spezifischen Korrektur- und Bewertungsbereich in der EASy-Plattform begutachten. Parallel zu den studentischen Lösungen werden dem Tutor auch die automatischen Korrekturen sowie ein Vorschlag für die Bewertung präsentiert. Der Tutor kann diese Systemvorschläge bestätigen, ergänzen oder vollständig durch eigene Beurteilungen ersetzen. Insbesondere einer manuellen Kommentierung der Leistungen und der Bereitstellung eines persönlichen Feedbacks



durch den Tutor wird didaktisch besonderer Wert beigemessen. Ist ein Beweis vollständig und korrekt, gibt das System eine textuelle und visuelle Rückmeldung, so dass der Tutor direkt die volle Punktzahl geben kann, ohne den Beweis im Detail nachvollziehen zu müssen. Wird ein Beweis nicht vollständig gelöst, muss der Tutor den Beweis manuell nachvollziehen und die angemessene Anzahl an Punkten notieren. Eine partielle Bewertung ist in der aktuellen Version des Aufgabenmoduls noch nicht realisiert wurde. Nach abgeschlossener Korrektur und Bewertung (oder ggf. auch erst nach einer Präsenzsitzung) wird dem Übungsteilnehmer das Feedback zu seiner persönlichen Leistung im System bereitgestellt. Die Tutoren und Dozenten können sich auf einer Übersichtsseite einen Überblick über den Erfolg der einzelnen Teilnehmer sowie der gesamten Gruppe verschaffen. Dies unterstützt sie bei der Vorbereitung von Präsenzübungen, bei der Erstellung von Statistiken und gibt Impulse, an welchen Stellen der Lehr-Lernprozess einer Anpassung bedarf.

EASy unterstützt folglich eine computerunterstützte Organisation des Übungsbetriebs, in der zwar nicht sämtliche Prozesse automatisch durch das System erfolgen und in der von Dozenten und Tutoren nach wie vor Initiative erwartet wird. Ihre Arbeitsbelastung wurde jedoch an didaktisch, methodisch und organisatorisch sinnvollen Stellen reduziert, wodurch zeitliche Ressourcen für die Ausweitung der direkten Betreuung der Studierenden freiwerden können.

3.2. Evaluation des Einsatzes von EASy

Es gibt wenige Erfahrungen über die Effektivität, Effizienz und Akzeptanz elektronisch unterstützter Übungen mit informatisch-mathematischem Schwerpunkt. Um die Prozesse generell zu verbessern und EASy langfristig zu etablieren, wurden auf Basis einer formativen Evaluation die Stärken und Schwächen des Systems ermittelt. Hierzu wurde EASy bereits im frühen Entwicklungsstatus im Rahmen der Bachelorveranstaltung *Datenstrukturen und Algorithmen* an der WWU Münster eingesetzt und evaluiert (Gruttmann et al., 2008a). Im Sommersemester 2008 wurde diese Veranstaltung von mehr als 250 Studierenden besucht. Neun Tutoren betreuten die Studierenden in Gruppen mit jeweils etwa 23 Teilnehmern.

Um sicherzustellen, dass EASy die Belange aller Nutzergruppen erfüllt, wurden sowohl die Tutoren als auch die Studierenden zu ihren Erfahrungen mit EASy befragt. In zwei nutzerspezifischen Umfragen wurde die Handhabbarkeit, Effektivität und Benutzerfreundlichkeit des Systems erforscht und die generelle Haltung der Nutzer gegenüber computerunterstützten Lernerfolgskontrollen ermittelt. Hierfür wurden Übungsaufgaben in EASy formuliert wie z. B. die Beweise folgender Aussagen:

- $\sum_{i=0}^n i = \frac{n \cdot (n+1)}{2}$
- Quicksort benötigt $O(n \cdot \log n)$ Schritte im Best-Case



Nachdem die Studierenden die Aufgaben bearbeitet hatten, wurden sie gebeten, einen Fragebogen auszufüllen, in dem sie ihre Erfahrungen und Kritik äußern konnten. Von den rund 250 Teilnehmern der Vorlesung nahmen 165 Studierende an der freiwilligen Evaluation teil ($n = 165$). Den Tutoren wurde ein ähnlicher Fragebogen im Anschluss an den Korrekturprozess ausgehändigt ($n = 9$). Zentrale Ergebnisse der Evaluation werden im Folgenden zusammengefasst.

Es zeigt sich, dass die meisten Studierenden die geforderten Aufgaben nach einer gewissen Einarbeitungszeit in EASy gut lösen konnten. Sowohl die Studierenden als auch die Tutoren waren sehr am Konzept des Systems interessiert und begrüßten einen weiteren Einsatz von EASy im Rahmen der Informatikübungen (siehe Abb. 7).

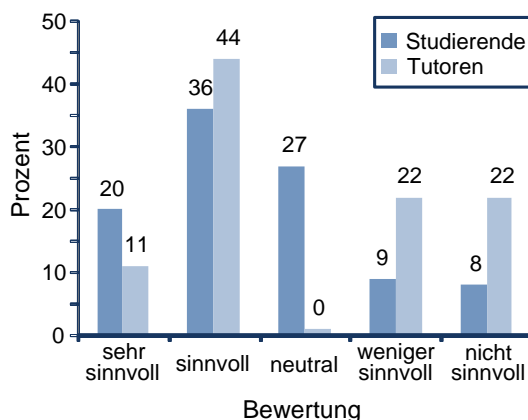


Abb. 4: Bewertung des Einsatzes von EASy im Übungsbetrieb.

Insbesondere die Tutoren profitierten vom Einsatz des E-Assessment-Systems. Sie gaben an, dass sich der Korrekturaufwand seit der Einführung von EASy signifikant verringert habe (Wert 1,9 auf einer Skala von 1 bis 5). Während im manuellen Modus die Korrektur einer Beweisaufgabe etwa 16 min. in Anspruch nahm, genügten nun (im Falle eines korrekten Beweises) wenige Sekunden, um festzustellen, dass die Aufgabe vollständig gelöst wurde, und um die volle Punktzahl zu notieren. Bei unvollständig abgegebenen Beweisen muss zwar nach wie vor manuell korrigiert werden, jedoch schätzten die Tutoren, dass die Beweise durch die elektronische Aufbereitung wesentlich klarer strukturiert und besser lesbar seien als handschriftliche Abgaben. Die Tutoren bewerten EASy und seine unterstützende Funktionalität zum Führen wohl strukturierter und korrekter Beweise insbesondere für Studierende mit weniger ausgeprägten mathematischen Kenntnissen als sehr hilfreich (44 %). Das manuelle Führen von Beweisen sei zwar durch EASy nicht zu ersetzen, die traditionellen Lernprozesse könnten jedoch sinnvoll ergänzt werden.

Während die Arbeitsbelastung bei den Korrekturen durch den Einsatz von EASy erheblich reduziert werden konnte, erhöhte sich der Aufwand bei der Erstellung der Lösungen. Als Gründe hierfür wurden die Einarbeitung in das System (11 % der Teilnehmer), die Auswahl von passenden Regeln aus der umfassenden Regelbasis (41 %),



die Kleinschrittigkeit, die EASy bei der Termumformung fordert (10 %) sowie die eigene mathematische Unsicherheit (12 %) angeführt. Dennoch lösten 78 % der Teilnehmer den Beweis vollständig mit Hilfe des Systems.

Die Studierenden profitieren jedoch auch von der elektronischen Unterstützung der Übungen. Die automatischen Korrekturen durch das System sind objektiv und konsistent, was die allgemeine Validität und Fairness der Bewertung sichert. Die Studierenden sahen das System als sehr hilfreich beim Erlernen allgemeiner Beweisstrategien an, da es die formalen Strukturen und potenzielle Lösungsschemata vorgibt (22 %). Die grafische und textuelle Visualisierung des Beweisstatus in EASy trug ihrer Meinung dazu bei, dass sie sich intensiver und länger mit der Lösung der Aufgaben beschäftigten, als sie es bei einem manuellen Beweis getan hätten. Dass Flüchtigkeitsfehler durch das System direkt abgeblockt und durch eine entsprechende Rückmeldung erläutert werden, begrüßen 19 % der Studierenden. Insbesondere in Grundlagenveranstaltungen wie *Datenstrukturen und Algorithmen* sei dies ein sinnvoller Nebeneffekt des Einsatzes von EASy.

Als Ergebnis der Evaluation ist festzuhalten, dass mit dem vorgestellten System EASy eine adäquate Unterstützung des vorlesungsbegleitenden Übungsbetriebs im Informatikstudium ermöglicht wird. Mit Hilfe des Aufgabenmoduls für mathematische Beweise können relevante analytische, kreative und konstruktive Fähigkeiten der Studierenden erfasst und bewertet werden. Weiterer Forschungsbedarf wird dabei in zweifacher Hinsicht gesehen. Zum einen sind im Bereich des E-Assessments im Generellen zusätzliche Arbeiten notwendig, um eine weitere Dissemination von E-Assessment-Systemen zu fördern. Zum andern sind spezifische Weiterentwicklungen von EASy denkbar, die z. B. auf die Unterstützung weiterer Aufgabentypen zielen, so dass der Anwendungskontext von EASy ausgebaut werden kann. Ferner werden eine Integration des EASy-Systems mit der vorhandenen E-Learning-Infrastruktur und eine Erweiterung um virtuelle Kollaborationsmöglichkeiten angestrebt.

4. Fazit

Der vorliegende Beitrag thematisiert das E-Assessment vorlesungsbegleitender Übungen im Informatikstudium. Es wurde untersucht, wie typische Prozesse der Vorbereitung, Durchführung und Nachbereitung der Übungsaufgaben durch neue Medien unterstützt werden können, um langfristig eine Entlastung von Lehrenden auf der einen Seite und einen effektiveren Übungsbetrieb für die Studierenden auf der anderen Seite zu erreichen. Anhand des E-Assessment-Systems EASy wurde aufgezeigt, wie der computerunterstützte Übungsbetrieb mit Hilfe eines innovativen Systems realisiert werden kann. Insbesondere für die computerunterstützte Durchführung und Bewertung mathematischer Beweise stellt EASy ein Novum dar.

Die Ergebnisse einer Evaluation des Praxiseinsatzes von EASy zeigen, dass sowohl Studierende als auch Tutoren vom entwickelten Konzept des computerunterstützten Übungsbetriebs profitieren und beide Nutzergruppen eine Fortführung des Konzepts



begrüßen. Gleichwohl wurden aber auch weitere Optimierungspotenziale hinsichtlich der Gestaltung von Lehr- und Lernprozessen mit EASy identifiziert. Diese Erkenntnisse der Evaluation fließen in die konkrete Konzeption zukünftiger Assessments ein und werden bei der Weiterentwicklung des Systems beachtet, um eine nachhaltige Etablierung von E-Assessment in der Hochschullehre zu erreichen. Es ist zu erwarten, dass durch den Einsatz von E-Assessment die Qualität des Übungsbetriebs entsprechender Lehrveranstaltungen an der WWU Münster nachhaltig gesteigert werden kann. Die gewonnenen Erkenntnisse lassen sich auf eine Vielzahl von weiteren Bildungseinrichtungen übertragen, da dort oftmals analoge Problemsituationen in mathematisch-technischen Lehrveranstaltungen vorzufinden sind.

Literaturverzeichnis

- ASIIN (2006), *Zur Akkreditierung von Bachelor- und Masterstudiengängen der Informatik - Fachspezifische ergänzende Hinweise*, in: (Hrsg.), Akkreditierungsagentur für Studiengänge der Ingenieurwissenschaften, der Informatik, der Naturwissenschaften und der Mathematik e.V., Düsseldorf 2006.
- Baader, F. & Nipkow, T. (1999), *Term Rewriting and All That*, Cambridge Univ. Press, Cambridge 1999.
- Chalmers, D. & McAusland, W. D. (2002), *Computer-assisted Assessment*, im WWW unter: <http://www.economicsnetwork.ac.uk/handbook/caa/>, [2009-11-24].
- Dyckhoff, A.; Rohde, P. & Stalljohann, P. (2008), An Integrated Web-based Exercise Module, in: V. Uskov (Hrsg.), *Proceedings of the 11th IASTED International Conference on Computers and Advanced Technologies in Education*, Crete 2008, S. 244-249.
- GI (2005), *Empfehlungen für Bachelor- und Masterprogramme im Studienfach Informatik an Hochschulen*, im WWW unter: http://www.gi-ev.de/fileadmin/redaktion/empfehlungen/GI-Empfehlung_BaMa2005.pdf, [2010-02-01].
- Görlitz, G. & Müller, S. (2002), Didaktisches Design für eine Online-Programmierausbildung, in: *Softwaretechnik-Trends*, 22 (2002) 3, S. 32-35.
- Gruttmann, S.; Böhm, D. & Kuchen, H. (2008a), E-Assessment of Mathematical Proofs – Chances and Challenges for Students and Tutors, *Proceedings of the 2008 International Conference on Information Technology in Education*, Wu-han 2008a.
- Gruttmann, S. & Kuchen, H. (2008), A Framework for Formative E-Assessments of Mathematical Proofs, *Proceedings of the Symposium on e-Assessment in Practice*, Shrivenham 2008.
- Gruttmann, S.; Kuchen, H. & Böhm, D. (2008b), An E-Assessment Systems for Mathematical Proofs, in: V. Uskov (Hrsg.), *Proceedings of the 11th IASTED International Conference on Computers and Advanced Technologies in Education*, Crete 2008b, S. 120-125.
- Hoare, C. A. R. (1969), An axiomatic basis for computer programming, in: *Communications of the ACM*, 12 (1969) 10, S. 576–585.



Kleimann, B. & Wannemacher, K. (2005), *E-Learning-Strategien deutscher Universitäten: Fallbeispiele aus der Hochschulpraxis*, Hannover 2005.

Thaller, G. E. (2000), *Software-Metriken einsetzen, bewerten, messen*, Technik Verlag, Berlin 2000.

Die Autorin / die Autoren



Susanne Gruttmann ist wissenschaftliche Mitarbeiterin am European Research Center for Information Systems (ERCIS) der Westfälischen Wilhelms-Universität Münster. Dort promovierte sie im Januar 2010 am Lehrstuhl für Praktische Informatik in der Wirtschaft zum Thema E-Assessment in der Hochschullehre und war federführend an der Entwicklung des E-Assessment-Systems EASy zur computerunterstützten Lernfortschrittskontrolle im Informatikstudium beteiligt. Weitere Forschungsschwerpunkte von Frau Gruttmann liegen in den Bereichen E-Collaboration und Usability von E-Learning-Systemen

URL: <http://www.wi.uni-muenster.de/pi/>

E-Mail: gruttmann@wi.uni-muenster.de



Herbert Kuchen ist Professor für Praktische Informatik in der Wirtschaft an der Westfälischen Wilhelms-Universität Münster. In Forschung und Lehre beschäftigt er sich mit den Gebieten Software Engineering, E-Learning sowie der deklarative Programmierung.

URL: <http://www.wi.uni-muenster.de/pi/>

E-Mail: kuchen@uni-muenster.de



Claus Alexander Usener ist wissenschaftlicher Mitarbeiter am Lehrstuhl für Praktische Informatik in der Wirtschaft am European Research Center for Information Systems der Westfälischen Wilhelms-Universität Münster. Im Rahmen seiner Diplomarbeit plante und entwickelte er für das System EASy ein Aufgabenmodul zur automatischen Überprüfung von Programmieraufgaben. Am Lehrstuhl für Praktische Informatik in der Wirtschaft ist er seit Ende 2009 im Rahmen seines Promotionsvorhabens im Bereich E-Assessment mit der Weiterentwicklung von EASy betraut.

URL: <http://www.wi.uni-muenster.de/pi/>

E-Mail: usener@wi.uni-muenster.de