

Commentarii
informaticae didacticae | 4

Dieter Engbring | Reinhard Keil |
Johannes Magenheim | Harald Selke (Hrsg.)

**HDI2010 – Tagungsband
der 4. Fachtagung zur
»Hochschuldidaktik
Informatik«**

Commentarii informaticae didacticae (CID)

Dieter Engbring | Reinhard Keil |
Johannes Magenheimer | Harald Selke (Hrsg.)

**HDI2010 – Tagungsband der 4. Fachtagung
zur „Hochschuldidaktik Informatik“**

9./10. Dezember 2010 in Paderborn

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der
Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind
im Internet über <http://dnb.d-nb.de/> abrufbar.

Universitätsverlag Potsdam 2010
<http://info.ub.uni-potsdam.de/verlag.htm>

Am Neuen Palais 10, 14469 Potsdam
Tel.: +49 (0)331 977 4623 / Fax: 3474
E-Mail: verlag@uni-potsdam.de

ISSN (print) 1868-0844
ISSN (online) 2191-1940

Die Schriftenreihe *Commentarii informaticae didacticae* (CID) wird
herausgegeben von:

Johannes Magenheimer, Universität Paderborn
Sigrid Schubert, Universität Siegen
Andreas Schwill, Universität Potsdam

Das Manuskript ist urheberrechtlich geschützt.

Online veröffentlicht auf dem Publikationsserver der Universität Potsdam
URL <http://pub.ub.uni-potsdam.de/volltexte/2010/4916/>
URN <urn:nbn:de:kobv:517-opus-49167>
<http://nbn-resolving.org/urn:nbn:de:kobv:517-opus-49167>

Zugleich gedruckt erschienen im Universitätsverlag Potsdam
ISBN 978-3-86956-100-4

Vorwort

Mit der 4. Tagung zur Hochschuldidaktik Informatik wird eine Reihe fortgesetzt, die ihren Anfang 1998 in Stuttgart unter der Überschrift „Informatik und Ausbildung“ genommen hat. Seither dienen diese Tagungen den Lehrenden im Bereich der Hochschulinformatik als Forum der Information und des Diskurses über aktuelle didaktische und bildungspolitische Entwicklungen im Bereich der Informatikausbildung. Aktuell zählen dazu insbesondere Fragen der Bildungsrelevanz informatischer Inhalte und der Herausforderung durch eine stärkere Kompetenzorientierung in der Informatik.

Die eingereichten Beiträge zur HDI 2010 in Paderborn veranschaulichen unterschiedliche Bemühungen, sich mit relevanten Problemen der Informatikdidaktik an Hochschulen in Deutschland (und z.T. auch im Ausland) auseinanderzusetzen. Aus der Breite des Spektrums der Einreichungen ergaben sich zugleich Probleme bei der Begutachtung. Letztlich konnten von den zahlreichen Einreichungen drei die Gutachter so überzeugen, dass sie uneingeschränkt in ihrer Langfassung akzeptiert wurden. Neun weitere Einreichungen waren trotz Kritik überwiegend positiv begutachtet worden, so dass wir diese als Kurzfassung bzw. Diskussionspapier in die Tagung aufgenommen haben.

Darüber hinaus konnten wir drei Personen gewinnen, die in eingeladenen Vorträgen ihre Sicht der hochschuldidaktischen Entwicklungen der letzten Jahre auf der Tagung darstellen. Dies sind in der Reihenfolge ihres Auftretens Arno Rolf (Universität Hamburg), der sich in seinem Beitrag mit dem Titel „Themengärten der Informatik-Ausbildung“ mit der Frage beschäftigt, wie man der Überforderung durch die Datenflut Herr werden kann. Uwe Kastens (Universität Paderborn) setzt sich in seinem Vortrag kritisch mit dem ‚Deutschen Qualifikationsrahmen‘ auseinander und Wilfried Hauenschild (Universität Paderborn) analysiert das Verhältnis von Allgemeiner Hochschuldidaktik und Fach-Hochschuldidaktik. Zum Vortrag von A. Rolf liegt ein Manuskript vor, mit dem wir diesen Band auch beginnen. Ansonsten ist dieser Tagungsband den aktuellen eingereichten Forschungsarbeiten zur Hochschuldidaktik der Informatik vorbehalten.

Der besondere Dank der Herausgeber dieses Tagungsbandes gilt dem Programmkomitee für die Begutachtung der Beiträge sowie den vielen hier nicht genannten Helfern für den Einsatz bei der Vorbereitung der Tagung und von anderen Aktivitäten in deren Umfeld.

Die Herausgeber

Paderborn, im November 2010

Programmkomitee:

Jörg Desel, KU Eichstätt
Peter Forbrig, Universität Rostock
Jörg Haake, FernUni Hagen
Andreas Harrer, KU Eichstätt
Reinhard Keil, Universität Paderborn (stellv. Vorsitz)
Ulrike Lucke, Universität Rostock
Johannes Magenheimer, Universität Paderborn (Vorsitz)
Thomas Ottmann, Universität Freiburg
Ulrik Schröder, RWTH Aachen
Sigrid Schubert, Universität Siegen
Carsten Schulte, FU Berlin
Andreas Schwill, Universität Potsdam
Silke Seehusen, FH Lübeck
Günter Siegel, Beuth Hochschule für Technik Berlin
Markus Steinert, TU München
Albert Zündorf, Universität Kassel
Olaf Zukunft, HAW Hamburg

Inhaltsverzeichnis:

Arno Rolf: Themengärten in der Informatik-Ausbildung	7
Marc Berges, Peter Hubwieser: Vorkurse in objektorientierter Programmierung: Lösungsansatz für einen leichteren Einstieg in die Informatik	13
Timo Kehrer, Udo Kelter: Eine aufwandsbeschränkte Einführung in die modellbasierte Softwareentwicklung	23
Ralf Romeike: Output statt Input – Zur Kompetenzformulierung in der Hochschullehre Informatik	35
Marcus Frenkel, Karsten Weicker: Pseudo: eine Programmiersprache auf der Basis von Pseudocode zur Unterstützung der akademischen Lehre	47
Wolfgang Reinhardt, Michael Mascher, Senol Gül, Johannes Magenheim: Integration eines Rapid Feedback Moduls in eine koaktive Lern- und Arbeitsumgebung	53
Stephan Raimer: Aquadrohne, Messdatenerfassung und Co. - Ein interdisziplinäres Projektmanagement als Teil des Wirtschaftsinformatikstudiums	59
Ira Diethelm, Malte Dünnebier: Ein virtueller Lernraum für die Informatiklehrerweiterbildung	65
Gabor Kiss: Analyse der Studienleistungen von Studierenden an der Universität Óbuda und deren Implikationen für die Informatikausbildung	71
Susanne Boll, Rolf Meinhardt, Sabine Gronewold, Larissa Krekeler: Migranten – Einführung eines neuen Studienprogramms an der Universität Oldenburg	79
Isa Jahnke, Tobias Haertel, Volker Mattick, Karsten Lettow: Was ist eine kreative Leistung Studierender? Erfahrungen eines kreativitätsförderlichen Lehrbeispiels	87
Jörg Hafer, Joachim Ludwig, Marlen Schumann: Fallstudien in medialen Räumen	93
Christoph Laroque, Jonas Schulte, Diana Urban: KoProV – Ein Lehransatz zur koordinierten Projektvorlesung auf Basis von Wissensmodulen	99

Themengärten in der Informatik-Ausbildung

Arno Rolf

Universität Hamburg
Vogt-Kölln-Straße 30
22527 Hamburg

E-Mail: rolf@informatik.uni-hamburg.de

Die Möglichkeiten sich zu informieren, am Leben der vielen Anderen teilzunehmen ist durch das Internet mit seinen Tweets, Google-Angeboten und sozialen Netzwerken wie Facebook ins Unermessliche gewachsen. Zugleich fühlen sich viele Nutzer überfordert und meinen, im Meer der Informationen zu ertrinken. So bekennt Frank Schirmmacher in seinem Buch *Payback*, dass er den geistigen Anforderungen unserer Zeit nicht mehr gewachsen ist. Sein Kopf komme nicht mehr mit. Er sei unkonzentriert, vergesslich und ständig abgelenkt.

Das, was vielen zum Problem geworden ist, sehen viele Studierende eher pragmatisch. Der Wissenserwerb in Zeiten von Internet und E-Learning läuft an Hochschulen häufig nach der Helene-Hegemann-Methode ab: Zunächst machen sich die Studierenden, z.B. im Rahmen einer Studien- oder Hausarbeit, bei Wikipedia „schlau“, ein Einstieg ist geschafft. Anschließend wird dieses Wissen mit Google angereichert. Damit ist Überblickswissen vorhanden. Mit geschickter copy-and-paste-Komposition lässt sich daraus schon ein „Werk“ erstellen. Der ein oder andere Studierende gibt sich mit diesem Wissenserwerb zufrieden und bricht seinen Lernprozess hier bereits ab.

Nun ist zwar am Ende jeder Studierende für seinen Wissenserwerb selbst verantwortlich. Die erkennbar unbefriedigende Situation sollte die Hochschulen aber herausfordern, das Internet in Vorlesungen und Seminaren auszuprobieren und sinnvolle Anwendungen zu entwickeln.

Beispiele gibt es durchaus. Unter der Metapher E-Learning hat sich ein umfangreicher Forschungsschwerpunkt an den Universitäten entwickelt. Einige Beispiele von vielen: So hat der Osnabrücker Informatik-Professor Oliver Vornberger seine Vorlesungen als Video ins Netz gestellt. Per RSS ist es möglich, Sequenzen aufs iPod zu laden. Die übliche Dozentenangst, dann würden sie ja vor leeren Bänken sitzen, scheint unbegründet. Sie werden von den Studierenden vor allem zur Prüfungsvorbereitung genutzt.

Wie ist das Internet, das für die junge Generation zu einem alles andere verdrängenden Universalmedium geworden ist, didaktisch in die Hochschullehre einzubinden? Wie also ist konkret mit diesen Herausforderungen umzugehen? Dies soll uns im Folgenden beschäftigen.

Eine Antwort auf die didaktische Herausforderung: Themengärten

Wie könnte eine zufriedenstellende Antwort auf Schirmmachers Klage aussehen, dass es im Netz zwar unendlich viel Müll, aber auch unzählige intelligente Gedanken gibt, die jedermann interessieren müssten. Wie also könnte ein Seminar aussehen, das diesen In-

formationspool nutzt und dabei die Spreu vom Weizen trennt? Wie könnte das Internet so eingebunden werden, dass Denken, Erkenntnis und Kreativität unterstützt werden? Widerstand dürfte dabei nicht von den Studierenden kommen, die über Blogs, Facebook etc. längst selbst Produzenten von Informationen sind und Plattformen wie Wikipedia, Google etc. zu wichtigen Bestandteilen ihrer Lernprozesse gemacht haben. Eher dürfte es die an Powerpoint gewöhnten Dozenten Überwindung kosten.

Ein interessanter Vorschlag kommt von außerhalb der Universitäten. Alexander Kluge schlägt vor, der digitalen Überforderung durch Verringerung der Datenmenge Herr zu werden und zwar so, „dass sie in Ihren Kopf passt, dann geht es doch... Wir brauchen Gärten zum Sammeln, Sortieren und Reduzieren der Datenflut“ (Interview mit Alexander Kluge, „Der Angriff der 13. Fee“, In: Freitag Nr. 52/53, 2009). Seine Idee könnte darauf hinauslaufen, z.B. Studierenden und Lehrenden in gemeinsamen Lernprozessen das Sammeln, Sortieren und Reduzieren der vielfältigen Netz-Informationen zu überlassen und so die Spreu vom Weizen zu trennen.

Dieses Konzept wurde am Fachbereich Informatik der Universität Hamburg im Seminar „Web 2.0-Design der Wissensgesellschaft“ erprobt. Die Studierenden hatten in Gruppen Thesen zu bearbeiten und dabei kontroverse Rollen einzunehmen. Thesen waren beispielsweise: „Die Ära des Gedruckten (Buch, Print, Newspaper) geht zu Ende. Internet, Handy und iPad sind die Totengräber. Gestalten Sie ein Modell für Magazine/Bücher im Jahre 2020 aus Sicht von Verlagen und Lesern.“ Oder: „Plattformkonzerne wie Google, Facebook, Apple & Co werden die globale Ökonomie unter sich aufteilen. Gestalten Sie eine Option zu dieser Zukunft 2020“.

Die Aufgabe der Studierenden bestand darin, im Netz Informationen und Argumente für ihre Positionen zu recherchieren, in Form von selbst definierten *Themengärten* zu strukturieren und diese mithilfe von Wikis oder Weblogs zu dokumentieren und zu kommunizieren. Gleichzeitig sind brauchbare Software-Plattformen bzw. -Werkzeuge zu recherchieren, auszuprobieren und einzusetzen.

In diesem Prozess lernen sie, die Spreu vom Weizen zu trennen, Texte zu bewerten und ihre Entscheidung zu rechtfertigen. Sie werden auch gefordert, sich mit längeren Texten auseinandersetzen, was für Facebook und Twitter sozialisierte Studierende zuweilen schon eine Herausforderung ist. Sie können Dossiers anlegen, die den Stand des Wissens für ein Thema mit Literaturhinweisen, Links, Blogs, Aufsätzen und eigenen Arbeiten und Kommentierungen abdecken.

Die Metapher Themengarten deutet an, dass die Studierenden entscheiden müssen, was dort wachsen und was schrumpfen sollte, was für ihren Lernprozess relevant ist und was nicht. Ein Garten ist nie fertig, er braucht Pflege und je nach Nutzer werden neue Pflanzen hinzukommen, verblühte müssen entsorgt werden.

Die beschriebene Architektur strebt ein hohes Maß an inhaltlicher Selbstorganisation und Kooperation an. Eigenarbeit soll sich mit Gruppenkommunikation verknüpfen, Orientierungskompetenz kann wachsen. Das Internet wird ansatzweise beherrschbar.

Ähnlich wie bei Plattformorganisationen, z.B. bei Google, erhöht jeder Beitrag die „Intelligenz“ und das Interesse am jeweiligen Themengarten. Je „schlauere“ dieser wird, desto besser sind die Such- und Lernergebnisse, was wiederum die Nutzung der Gärten verstärkt. Denkbar und auch wünschenswert, dass in Seminaren auf diese Weise dem Regime der Powerpoint präsentierenden Studierenden und Hochschullehrer das Licht ausgeblasen werden kann.

Bewertung der Themengärten durch die Teilnehmer des Seminars

Die Studierenden wurden gebeten, neben der inhaltlichen Arbeit an ihrem Thema das Seminar schriftlich zu bewerten und dabei auch einen Vergleich mit üblichen Seminarformen vorzunehmen. Typisch für die traditionellen Seminare sind: Verteilung der Themen zu Beginn des Semesters sowie verpflichtend für alle ein Vortrag in der Regel in Form einer Powerpoint-Präsentation. Im Folgenden einige Auszüge der Teilnehmer:

S.H & D.K.: Bei dieser Form des Seminars hatten wir das Gefühl, dass alle gleichermaßen in das Seminar eingebunden waren und sich weitaus mehr Leute an den Diskussionen beteiligt haben, als in anderen Seminaren. Dadurch haben wir auch die Themen anderer Gruppen gut verfolgen können und haben nicht nur etwas über unser eigenes Thema gelernt. ... Im Gegensatz dazu vergisst man bei der üblichen Seminarform das meiste des Gelernten, nachdem man sein Referat gehalten hat, da man sich dort meist nur kurz und unmittelbar vor dem Referat mit seinem Thema beschäftigt.

J.B.: Ich habe den Eindruck gewonnen, dass das neue Konzept sehr viel Potential bietet, sowohl in methodischer Hinsicht als auch in Bezug auf die Ergebnisse. Allem Anschein nach ... ist allerdings die Dokumentation des Lernprozesses, die ja das Hauptanliegen war, überall bestenfalls auf ein Beiwerk reduziert worden. ... Ich befürworte unbedingt diese Seminarform (plädiere aber dafür) künftig potentielle Teilnehmer schon im Kommentierten Vorlesungsverzeichnis auf den Zeitbedarf hinzuweisen, sodass sie andere Veranstaltungen im selben Semester reduzieren.

J.B.: Was die Beurteilung der Seminarform im Gegensatz zur üblichen Lehrveranstaltung betrifft, muss ich ganz klar sagen, dass ich diese Form nicht erwartet habe und es daher umso „erfrischender“ fand, einmal einen anderen Weg einzuschlagen. Nachdem ich im vorherigen Semester das Proseminar besucht habe, bin ich bei dem Seminar von einer identischen Lehrveranstaltung ausgegangen. Überspitzt gesagt: möglichst früh das zugewiesene Thema bearbeiten und präsentieren, um die restlichen Vorträge dann mehr oder weniger „abzusitzen“. Dazu dann eine Standard-Haularbeit. ... Durch die frühzeitige Ankündigung als Experiment konnte man die Themen und deren gesamte Bearbeitung ganz anders angehen und wesentlich flexibler und kreativer handhaben. Dies äußerte sich beispielsweise darin, dass wir in unserer Präsentation bewusst auf eine Power-Point-Präsentation verzichtet haben, ein Novum bei Vorträgen in meiner bisherigen Schul- und Unilaufbahn. Es wäre also wünschenswert, wenn sich diese Seminarform durchsetzen könnte, denn von der üblichen Form mit Referat und Hausarbeit haben nun wirklich alle genug.

J.S.: Meiner Meinung nach sollte man beim nächsten Versuch den Leuten bereits eine modifizierte Version von Wordpress geben und anhand von Beispielen darlegen, wie man diese Software am besten nutzen könnte. ... Diese sollte dann dazu dienen, mit den anderen Gruppen zu kommunizieren. Also man lässt jede Gruppe in die Diskussionen der anderen Gruppen und veranlasst diese dazu, Fragen zu stellen.

T.K.: Findet die Kommunikation via „Web“ statt, sollte sich die Gruppe als solche kennengelernt haben, um eine gemeinsame Basis für die weitere Arbeit zu schaffen. ... Die Anonymität führte dazu, dass die individuelle, als auch die wechselseitige Verantwor-

tung der Teilnehmer in dieser Form des sogenannten kooperativen Lernens nicht wahrgenommen wurde. Der Blog unserer Gruppe ist mit der Zeit schlichtweg „eingeschlafen“. ... Erst durch ein Treffen der Arbeitsgruppe ist er wieder zu neuem Leben erwacht und es fand wieder eine intensivere Auseinandersetzung mit dem Blog statt. ... Obwohl die im Seminar eingesetzten Mittel ein Arbeiten von allen Orten und zu allen Tageszeiten ermöglichten, scheinen die persönlichen Treffen wichtiger zu sein, als in einem „normalen“ Seminar. ... Die Kommunikation durch Web 2.0 Anwendungen ist immer nur eingeschränkt möglich. Der Bereich der nonverbalen Kommunikation bleibt unberücksichtigt. ... Der Aspekt der Beziehungsebene und der damit verbundenen Emotionen kann nur unzureichend berücksichtigt werden. Daraus ergibt sich die Frage: Kann ein Seminar überhaupt ausschließlich über Web-2.0-Anwendungen funktionieren? Meine Erfahrungen in diesem Seminar haben gezeigt, dass diese Anwendungen nur unterstützend zur Seite stehen können. Blogs, Wikis und Mahara bilden aber einen sehr hilfreichen und der heutigen Zeit angemessenen Rahmen. ... Zum Bereich der Motivation bleibt anzumerken, dass meine eigene Motivation zu Beginn des Seminars am höchsten war. Ich habe Artikel gesammelt und gebloggt. Leider musste ich mit der Zeit feststellen, dass die eigene Motivationskurve absinkt, wenn man nicht die gewünschten Reaktionen auf die Artikel erfährt. Dies war des öfteren der Fall, wodurch unser Blog eher zu einer Art Materialsammlung wurde, anstatt zu einer offenen Diskussionsplattform. Wann immer ich Medien konsumiert habe, hatte ich zugleich ein offenes Auge für unseren Themenbereich. Das Seminar war in dem Punkt also sehr erfolgreich, dass es mich dazu gebracht hat, mich mit dem Thema des Arbeitsplatzes der Zukunft auseinander zu setzen und darüber zu informieren. Ich habe zum Beispiel viel über Co-Working Spaces, RFID-Chips und deren Auswirkung auf die Arbeitswelt gelernt. Auch Freelancer sind nun kein Fremdbegriff mehr für mich. All dieses Wissen hat sich ohne stupides Auswendiglernen in meinem Langzeitgedächtnis verankert. Das „Lernen“ in diesem Seminar basierte größtenteils auf intrinsischer (also eigener, von innen heraus) Motivation. Ich persönlich empfand dies als sehr förderlich, um Wissen zu erwerben, das man weit über das Seminar hinaus verwenden kann.

O.P.: Allerdings bezweifelte ich aber auch gleich die Realisierbarkeit des neuen Konzepts, weil ein grundlegendes Problem, das zu dem derzeitigen Verhalten der Studenten geführt hat, nämlich Zeitmangel, allein durch die Ablaufänderung einer Veranstaltung im Semester, nicht gelöst wird ... Unabdinglich bleibt aber dennoch, dass sich alle Teilnehmer über die Aktivitäten aller anderen Gruppen auf dem laufenden halten, damit sie in den Diskussionen in den Sitzungen auch über die neu gewonnenen Erkenntnisse in diesen Gruppen im Bilde sind.

Falls es in einer Gruppe kaum Diskussionen und neue Erkenntnisse gibt, liegt es in der Verantwortung des Kursleiters, durch neue Impulse und Anregungen, die Diskussion wieder ins Rollen zu bringen. Generell ist die Teilnahme der Veranstalter an den Diskussionen nicht nur in den Sitzungen, sondern auch in den Blogs von Vorteil, da dies den Studenten das Interesse der Veranstalter an den Diskussionen zeigt und dadurch zusätzlich motiviert werden.

F.B.: Die meiste Arbeit verlief dann jeweils in Eigenarbeit. Jeder recherchierte für sich im Netz und dokumentierte seine Funde dann im Blog. Dabei war es gar nicht so leicht, die große Menge an Informationen zu filtern, um möglichst nur seriöse Quellen zu nutzen. ... Zu Beginn des Seminars sollte bereits über die möglichen Systeme zur Umsetzung berichtet werden oder sich direkt auf ein einheitliches System geeinigt werden. Dadurch

bliebe mehr Zeit zur Erarbeitung der Ergebnisse. Bei einem einheitlichen System wäre zudem die Kommunikation mit anderen Gruppen leichter möglich.

N.L.: Von Anfang an stellte die Kommunikation in unserer Gruppe keine große Barriere dar. Wir kannten uns alle untereinander bereits und konnten daher konstruktiv miteinander arbeiten, ohne viel Zeit mit unsachdienlichen Diskussionen zu verschwenden. ... Auch muss den Gruppen deutlich gemacht werden, dass ein Prozess zu dokumentieren ist und nicht einfach die Recherche und Materialsicht ins Internet verlagert werden soll. Auch für internetaffine Studierende muss klar sein, dass die Diskussion als Mittel zum Wissenserwerb im Zentrum steht. ... Abschließend bin ich der Meinung, dass diese neue Seminarform, natürlich aktuell noch mit Schwächen, eine sehr positive Alternative zur aktuellen Form ist. Sie fördert und fordert die Studierenden eher und lässt sich dennoch akzeptabel in die aktuell, mit Zeitmangel behafteten, vorherrschenden Studienpläne integrieren. Auch ist die Bildung von wirklichem Interesse bei den Studierenden schneller zu erreichen.

M.R.: Was zu Beginn schwierig war, war die unterschiedliche Herangehensweise an das Medium Weblog. Ich denke, man hätte zu Beginn des Seminars eine Einführung in das Medium geben sollen. ... Blogs betreibt man aus einem bestimmten Bedürfnis heraus. Es ist eine bestimmte Einstellung, die man haben muss. Man muss das Bedürfnis haben etwas mit anderen Menschen zu teilen, sich mit anderen Menschen in Austausch zu begeben. Ein Blog ist mehr als nur ein Dokumentations-Werkzeug, für mich ist es zunächst ein Kommunikations-Medium. Ein Blog ist eine Möglichkeit, mit anderen Menschen in Kontakt zu geraten und über ein bestimmtes Thema zu kommunizieren. ... Um einen stetigen Austausch zu sichern, sollte jeder Teilnehmer pro Woche zwei Blogbeiträge anfertigen und drei weitere kommentieren müssen. ... Wenn man sich besser gekannt hätte, hätte ich mir vorstellen können, auch außerhalb des Seminars mit den anderen in Kontakt zu geraten, wie zum Beispiel mit der Hilfe von Skype.

F.N.: Ursache für den unzureichenden Gruppen-Lernprozess war mit großer Wahrscheinlichkeit die fehlende persönliche Kommunikation. Nach jedem Treffen im Plenum, bei dem sich unsere Gruppe real zusammengefunden hatte, wurden wir wieder motiviert, unseren Blog zu erweitern und Beiträge zu posten. Die persönliche Kommunikation begünstigt den Lernfortschritt also erheblich.... mussten wir am Ende alle feststellen, dass uns die Arbeit mit dem Blog Freude bereitet hat. Es fand kein stures Lernen statt, sondern der Lernprozess fand unbewusst statt. ... es hat sich gezeigt, dass trotz aller Kommunikationsmöglichkeiten durch IT die persönliche Kommunikation nicht unterbleiben darf und ihr immer noch ein hoher Stellenwert zukommt.

C.M.: Durch diese neue Seminarform ist es zum einen möglich, den Lernprozess abzubilden und nicht nur das Ergebnis, andererseits bringt das Lernen nach meiner Erfahrung so mehr Spaß. Man ist freier in seiner Materialauswahl, kann sich die Quellen zusammensuchen, die einen wirklich interessieren und ist nicht gezwungen, zu einem festgelegten Ergebnis zu kommen. Ich denke, dass ich das, was ich über mein Thema gelernt habe, länger im Gedächtnis behalten werde, da ich mit Interesse daran gearbeitet habe. Mir ist auch sehr positiv aufgefallen, dass es während der Treffen zu regen Diskussionen kam, was ich bisher in einem Seminar noch nie erlebt hatte, was wohl daran liegen mag, dass monoton gehaltene Referate nicht besonders diskussionsförderlich sind. Auch den sekundären Einsatz von Powerpointfolien befürworte ich. Dadurch wurden die Vorträge

lebendiger und man konzentrierte sich mehr auf die Personen, anstatt mit den Augen an der Präsentation „zu kleben“. Natürlich ist es für den Vortragenden zu Beginn schwer, sich damit zu arrangieren, da man plötzlich frei reden muss, ohne eine Präsentation zu haben, an der man sich orientieren kann. Aber genau das sollte meiner Meinung nach auch gefördert werden. Es ist nur eine Frage der Übung.

M.S.: Um jedoch die Beiträge den Kategorien zuzuordnen, gab es persönliche Treffen, bei denen interessante Diskussionen entstanden, die zu einem kontroversen gemeinschaftlichen Lernprozess bei allen Gruppenmitgliedern führten. Dieser Prozess hätte ebenfalls im Blog stattfinden können, jedoch ist eine synchrone reale Diskussion für uns ein großer Mehrwert im Gegensatz zu einer virtuellen asynchronen Diskussion und trägt aktiver zur Wissensbildung bei, bzw. hat einen hohen Stellenwert für den Lernprozess. Fazit hieraus ist, dass für den Meinungsbildungsprozess und den Lernprozess auch mit dem Hintergrund des Web 2.0 persönliche Treffen unerlässlich sind und bleiben werden. ... Dieses Seminar hat sehr interessante und innovative Lehr- & Lernmethoden ins Leben gerufen, die hoffentlich bald von weiteren Dozenten umgesetzt werden. Sie ist nicht nur in den Bereichen Sozialkompetenz, Medienkompetenz und Urteilsvermögen sehr viel förderlicher als die ursprüngliche Form des Seminars, sondern macht Spaß und gibt einen weiten Blick über den Tellerrand der „Suppe des Faktenlernens“.

Die exemplarische Umsetzung des Konzeptes Themengärten wurde am Studienschwerpunkt „Informatiksysteme in Organisationen und in der globalen Gesellschaft“ der Universität Hamburg, Fachbereich Informatik realisiert.

Eine abschließende Anmerkung: Die Absicht dieses Beitrages ist es nicht, über ein empirisch abgesichertes Forschungsprojekt zu berichten. Vielmehr ging es uns darum, Dinge zu erproben. Die Ergebnisse sind jedoch so ermutigend, dass wir uns wünschen, dass dieses Experiment auch an anderen Hochschulen erprobt und darüber berichtet wird.

Literatur

- [K09] Kluge, Alexander: Der Angriff der 13. Fee, In: Freitag Nr. 52/53, 2009.
- [KRCS06] Krause, D., Rolf, A., Christ, M., Simon, E.: Wissen, Wie alles zusammenhängt – Das Mikropolis-Modell als Orientierungswerkzeug für die Gestaltung von Informationstechnik in Organisationen und Gesellschaft. In: Informatik Spektrum Heft 4, S.263-273, 2006.
- [R08] Rolf, A.: MIKROPOLIS 2010. Menschen, Computer, Internet in der globalen Gesellschaft, Metropolis Verlag, Marburg 2008.
- [S09] Schirmacher, Frank: *Payback*, Karl Blessing Verlag, München 2009.

Vorkurse in objektorientierter Programmierung: Lösungsansatz für einen leichteren Einstieg in die Informatik

Marc Berges

Fachgebiet Didaktik der Informatik
TU München
85748 Garching - Germany
Web: <http://www.ddi.tum.de>
Email: berges@tum.de

Peter Hubwieser

Fachgebiet Didaktik der Informatik
TU München
85748 Garching - Germany
Web: <http://www.ddi.tum.de>
Email: peter.hubwieser@tum.de

Zusammenfassung: Die Studienanfänger der Informatik haben in Deutschland sehr unterschiedliche Grundkenntnisse in der Programmierung. Dies führt immer wieder zu Schwierigkeiten in der Ausrichtung der Einführungsveranstaltungen. An der TU München wird seit dem Wintersemester 2008/2009 nun eine neue Art von Vorkursen angeboten. In nur 2,5 Tagen erstellen die Teilnehmer ein kleines objektorientiertes Programm. Dabei arbeiten sie weitestgehend alleine, unterstützt von einem studentischen Tutor. In dieser Arbeit sollen nun das Konzept der sogenannten „Vorprojekte“ sowie erste Forschungsansätze vorgestellt werden.

1 Einleitung

Die Studenten, die jedes Jahr an den deutschen Hochschulen ihr Informatikstudium beginnen, starten mit sehr unterschiedlichen Kenntnissen in der Programmierung. Insbesondere das Wissen in der objektorientierten Programmierung (OOP) differiert sehr stark. Dies führt immer wieder zu großen Schwierigkeiten in den Einführungsveranstaltungen des ersten Semesters. Einige Studenten verfügen bereits über Programmierkenntnisse in imperativen Sprachen und müssen sich an das objektorientierte Paradigma erst gewöhnen. Andere kommen komplett ohne Vorkenntnisse und müssen zu den vielen neuen Konzepten, die vermittelt werden, auch noch eine Programmiersprache erlernen. An der Technischen Universität München wird daher seit dem Wintersemester 2008/2009 ein völlig neuer Weg beschritten. Vor dem Semesterbeginn wird ein freiwilliger Vorkurs für Studienanfänger angeboten. Unter der Leitung eines studentischen Tutors lernen die zukünftigen Studenten in sehr kurzer Zeit die einfachsten Grundlagen der OOP, ohne dabei einen Vortrag zu hören. Der Fokus liegt unter anderem auf der Überwindung der Angst vor dem selbstständigen Programmieren.

2 Beschreibung der Vorprojekte

Das in der Einleitung beschriebene Problem der Heterogenität der Studienanfänger bezüglich der Programmierkenntnisse ist auch an der TU München stark vorhanden. Im Zuge der Einführung der Studienbeiträge wurden vom Fachgebiet Didaktik der Informatik Vorkurse in objektorientierter Programmierung eingeführt.

Die Frage, wann und in welchem Maße objektorientierte Programmierung Inhalt einer Einführungsveranstaltung in der Informatik sein muss, wird in diversen Arbeiten ausgiebig behandelt [WH1, DT1]. Um möglichst früh in die theoretischen Konzepte einzusteigen, ist ein gewisses Handwerkszeug nötig, damit diese Konzepte auch praktisch erprobt werden können. Diese Werkzeuge können nicht im Rahmen der Einführungsveranstaltung in die Informatik vermittelt werden.

Das Institut für Informatik hat deswegen bereits der Einführungsveranstaltung ein Praktikum angegliedert, in dem die Konzepte praktisch erprobt werden können. Damit dieses Praktikum nicht zu einem bloßen Programmierkurs verkommt, ist es nötig, einige Programmierkenntnisse in das Studium mitzubringen. Dies wiederum soll aber kein Aufnahmekriterium für das Informatikstudium sein. Es muss also sichergestellt werden, dass einfaches Programmieren von den Studienanfängern beherrscht wird, was den Rahmen für die Vorprojekte bildet.

Ein weiteres Problem ist die Überschneidung solcher Vorkurse mit der jeweiligen Einführungsveranstaltung. Auch in unseren Vorprojekten sollten keine wesentlichen Inhalte vorweg genommen werden. Der erste Schritt in diese Richtung wurde bereits mit der Konzeption der Arbeitsblätter übernommen. Zum zweiten wurden die Veranstaltungen auf 2,5 Tage reduziert. Wie sich herausstellte, war das genug Zeit, um erste Programmieraufgaben zu lösen, aber sicherlich zu wenig, um ganze Konzepte zu vermitteln.

Das Konzept der Vorprojekte war von Beginn an mehrschichtig aufgebaut. Zum einen soll den beginnenden Studenten die Angst vor dem selbstständigen Lernen und Arbeiten genommen werden, zum anderen sollen die Studenten möglichst mit einem Grundlevel an Programmierkenntnissen in das Studium starten. Um nicht eine komplette Einführungsveranstaltung zu ersetzen, sollen die Studienanfänger nur die wirklich notwendigsten Konzepte gezeigt bekommen. Das selbstständige Ausprobieren steht ganz zentral im Mittelpunkt. Das besondere an den Vorprojekten ist, dass die angehenden Studenten keinen Vortrag zu hören bekommen. Ihnen werden lediglich vier Arbeitsblätter¹ gegeben, mit denen sie sich ihre ersten Programmierkenntnisse erwerben sollen. Damit hat jeder die Chance, seinen individuellen Lernweg zu beschreiten.

Als ganz zentrales Element wird jeder Gruppe, die aus acht bis zehn Studenten besteht, ein studentischer Tutor zugewiesen, der für Fragen zur Verfügung steht. Diese werden zumeist im Rahmen des etablierten Tutorenbetriebs am Institut für Informatik der TU München angeworben. Dies hat entscheidende Vorteile. Diese Tutoren haben eine didaktische Schulung bekommen [SB1] und verfügen über einige Erfahrung in der Lehre, da sie in den Basisvorlesungen eingesetzt werden. Dadurch sind sie mit den Problemen der Studienanfänger sehr vertraut. Außerdem hat sich der enge Kontakt der angehenden Studenten zu erfahrenen Studenten als sehr wichtig erwiesen.

Ein weiteres zentrales Element ist die Individualität des Lernweges. Durch die unterschiedlichen Vorkenntnisse ist eine möglichst individuelle Betreuung unbedingt notwendig. Diese fördert das selbstständige Umsetzen des Gelernten und ermöglicht so ein optimales Lernen. In der „Cognitive Load Theory“, u.a. vorgestellt in [CB1], wird davon ausgegangen, dass effektives Lernen nur dann möglich ist, wenn die Kapazität des Arbeitsgedächtnisses möglichst optimal ausgenutzt wird. Dabei ist es vor allem wichtig, dass die Elemente Muster aus ähnlichen Inhalten bilden. Das praktische „Ausprobieren“, das in den Vorprojekten ganz besonders gefördert wird, unterstützt diese Musterbildung und vereinfacht so die Übernahme ins Langzeitgedächtnis.

¹ <http://vmhub1.informatik.tu-muenchen.de/publikationen/material/vorprojekte.pdf>

Damit die individuelle Hilfe auch auf optimalem Niveau erfolgen kann ist es nötig, möglichst homogene Gruppen zu bilden. Dazu werden die Teilnehmer bei der Anmeldung um eine Selbsteinschätzung ihrer Programmierkenntnisse gebeten.

Es kann zwischen folgenden Kategorien gewählt werden:

- 1 – keine Programmierkenntnisse
- 2 – schon mal programmiert
- 3 – schon mal objektorientiert programmiert

2.1 Konkrete Zahlen der ersten beiden Durchgänge

In Tabelle 1 sind die Anmeldezahlen, sowie die Verteilung auf die „Kenntnis“-Gruppen tabellarisch aufgelistet. Außerdem wird der Anteil der Teilnehmer an der Anzahl der Studienanfänger in den Bachelor-Studiengängen Informatik und Wirtschaftsinformatik dargestellt. Der Rückgang der Anmeldungen ist auf zwei Gründe zurückzuführen. Erstens sind die Studierendenzahlen leicht gesunken. Außerdem wurden die Studienanfänger der Bioinformatik nicht mehr explizit angeschrieben, da die Einladung im Rahmen der Immatrikulation an der TU München stattfindet. Die Bioinformatiker sind aber an der LMU München eingeschrieben. Trotzdem haben sich weniger Teilnehmer als im ersten Jahr angemeldet.

Tabelle 1: Aufschlüsselung der Teilnehmerzahlen

Semester	Anmeldungen	Gruppe 1	Gruppe 2	Gruppe 3	Anteil an Studienanfängern
WS08/09	200	114	42	44	53%
WS09/10	135	80	30	25	39%

3 Die Arbeitsblätter

Da in den Vorprojekten auf Vorträge jeglicher Art verzichtet wird, bekommen die verteilten Arbeitsmaterialien und die Tutoren eine noch gewichtigere Rolle. Auf das Tutoriensystem wurde bereits im vorherigen Abschnitt ausführlich eingegangen. In diesem Abschnitt sollen nun die Arbeitsblätter näher betrachtet werden. Zunächst müssen allerdings einige theoretische Grundlagen geklärt werden.

3.1 Programmieren

Damit der Rahmen der Arbeitsblätter festgelegt werden kann, muss zunächst geklärt werden, was Programmieren lernen eigentlich heißt. Nach Boyer, Langevin und Gaspar heißt programmieren lehren, die Studenten zu befähigen, die Anforderungen zu analysieren und die benötigten Algorithmen zu ermitteln. Diese müssen dann in einer Programmiersprache umgesetzt und auf Korrektheit überprüft werden können [BLG1]. Ordnet man also die Kompetenz des objektorientierten Programmierens in die Taxonomie von

Anderson/Krathwohl [AK1] ein, so lassen sich die kognitiven Elemente der Wissensdimension in die Kategorie Factual Knowledge einordnen [BLG1].

3.2 Minimalkonzept der Objektorientierung

Sucht man nach Definitionen zur Objektorientierung findet man eine kaum zu überblickende Anzahl. Es gibt die verschiedensten Herangehensweisen. Henrik Christensen beschreibt in [HC1] das Problem der verschiedenen Definitionen und Herangehensweisen sehr genau. Hier werden die gefundenen Definitionen in verschiedene Perspektiven eingeteilt. Die „language centric perspective“ stellt die Klasse und das Objekt als Elemente zur Konstruktion von Software in den Mittelpunkt. Die „model centric perspective“ sieht Objekte immer im Zusammenhang mit einem Modell. Die letzte Perspektive, die „responsible centric perspective“ stellt die Interaktion der Objekte in den Vordergrund. Der Kern der meisten Definitionen bleibt aber der gleiche. Hier soll zunächst die Definition von Hares [HS1] exemplarisch genannt werden:

„An object is, in terms, composed of two types of information, the data and logic information components. An object does not have to have both types of information simultaneously, and can contain data and logic or just logic. The object must contain logic, because, as we shall see, the data component cannot be accessed except via the logic component. The data component is optional.“

Aus den verschiedenen Definitionen haben wir die wichtigsten Kernelemente herausgenommen. Ein Programm, welches der OOP entspricht, muss aus mindestens einer Klasse bestehen. Zudem müssen Objekte instanziiert werden. Ein weiterer sehr wichtiger Aspekt ist die Datenkapselung. Daraus resultiert, dass zumindest eine Methode in der Klasse enthalten sein muss.

Für die Konzeption unserer Arbeitsblätter haben wir uns an unserem Minimalkonzept und der „language centric perspective“ orientiert. Diese beiden Elemente bilden mit den nötigen Elementen der Programmiersprache Java die Inhalte der vier Arbeitsblätter, die an die Studenten verteilt werden. Im nächsten Abschnitt werden die Arbeitsblätter nun im Detail beschrieben.

3.3 Aufbau der Arbeitsblätter

Die benötigte Information für die Programmierprojekte wird auf vier Arbeitsblätter komprimiert. Die vier Elemente des Programmierenlehrens aus [BLG1], nämlich „Anforderungsanalyse“, „Algorithmenermittlung“, „Implementierung“ und „Prüfung auf Korrektheit“, geben der Rahmen der Arbeitsblätter vor. Die Prüfung auf Korrektheit der jeweiligen Implementierungen wird allerdings nicht explizit thematisiert, da die Zeit der Vorkurse zu knapp ist. Außerdem würde dies zu sehr dem Inhalt der Einführungsveranstaltung vorweggreifen.

Auf dem ersten Arbeitsblatt werden die groben Ideen hinter der Objektorientierung erklärt. Dabei liegt der Fokus im Verstehen des Objektbegriffs. Attribute und Methoden bilden die zentralen Inhalte. Konzepte wie Vererbung und Polymorphie werden gänzlich ausgelassen. Diese würden ebenso den Inhalten der Einführungsveranstaltung zu stark vorweggreifen und den engen Zeitrahmen sprengen.

Nach dem theoretischen Teil über Objekte sollen die Teilnehmer ihre frisch erworbenen Kenntnisse anhand des Programms ObjektDraw einsetzen. ObjectDraw wurde von Martin Papst zur Unterstützung der Einführung in die Objektorientierung an der Schule entwickelt. Hier können Grafiken als Objekte erzeugt werden und durch Methodenaufrufe

beeinflusst werden. Desweiteren werden alle Objekte durch Objektkarten dargestellt und können so sehr gut beobachtet werden.

Als Aufgabe sollen die Teilnehmer ein Fußballfeld oder ein anderes Sportfeld grafisch darstellen. Dabei sollen immer wieder die Veränderungen der Objekte beobachtet werden. Das erste Blatt beinhaltet außerdem die Aufgabentexte zu den drei Schwierigkeitsgraden.

Auf dem zweiten Blatt steht die Einführung in die Entwicklungsumgebung BlueJ im Vordergrund. BlueJ bietet einen einfachen visuellen Zugang zu den Objekten [BBK1]. Bei den Aufgaben auf Blatt 2 haben wir uns an den Vorschlägen von Kölling und Rosenberg aus [KR1] orientiert. In einem ersten Schritt sollen die Teilnehmer den Umgang mit BlueJ anhand des Beispielprojekts Shapes erproben. Danach sollen sie aus dem Aufgabentext die Klassen, Attribute und Methoden extrahieren und in BlueJ anlegen. Um die Orientierung im Quellcode zu vereinfachen, ist dem zweiten Blatt eine Beispielklasse angefügt.

Im dritten Blatt folgt nun der Einstieg in die Java-Programmierung. Dazu sind auf dem Blatt die wichtigsten Java-Konstrukte erklärt. Außerdem wird die Syntax grafisch dargestellt (siehe Abbildung 1). Die Darstellung entstammt [MW1]. Mit den Elementen auf dem dritten Blatt lassen sich bereits die Methoden und Attribute der Klassen anlegen.

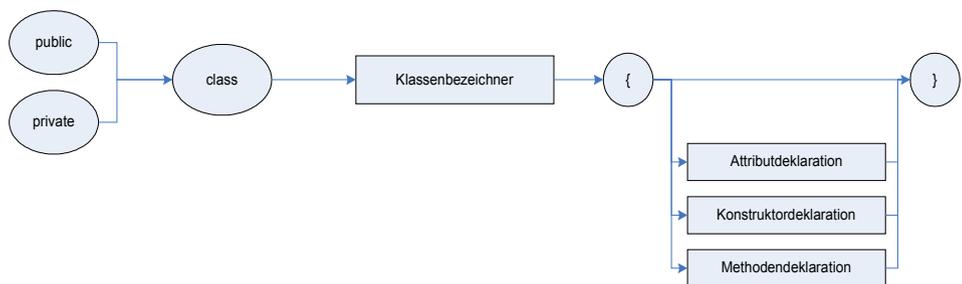


Abbildung 1: Grafische Darstellung eines Java-Konstrukts (class)

Das Füllen der Methodenrumpfe ist Inhalt des vierten Arbeitsblattes. Zu Beginn werden den Teilnehmern zwei Möglichkeiten gezeigt, Algorithmen zu modellieren. Danach werden die fehlenden Java-Konstrukte erklärt. Damit haben die Teilnehmer alle wichtigen Sprachelemente zusammengefasst dargestellt, um die Aufgabe zu lösen.

4 Einschätzung der Teilnehmer

Neben der Qualität des Codes, der von den Teilnehmern erstellt wurde, ist die Selbsteinschätzung von großer Bedeutung. Eine der zentralen Aufgaben der Vorprojekte ist es, die Angst vor dem selbstständigen Programmieren zu nehmen. Ein weiteres Ziel, das wir erreichen wollen, ist, dass die angehenden Studenten sehr früh daran gewöhnt werden, sich weiterführende Informationen selbstständig zu besorgen. Dafür muss ihnen natürlich der Weg dorthin gezeigt werden. Dies wird durch die individuelle Betreuung mit Hilfe der Tutoren gewährleistet.

Die Fragebögen sind in sechs Teile aufgeteilt und wurden anonym ausgefüllt. Im Folgenden sollen die einzelnen Teile näher beschrieben und ausgewertet werden. Die dargestellten Ergebnisse beziehen sich dabei auf den Durchgang 2009/2010.

4.1 Personendaten

Der Fragebogen erfasst in seinem ersten Abschnitt Personendaten wie Alter und Geschlecht. Besonders das Geschlechtsmerkmal ist zur Untersuchung unter Gender-Aspekten sehr wertvoll.

In einer ersten Untersuchung wurde geprüft, ob der Anteil der weiblichen Teilnehmer sich vom Anteil an den Studienanfängern unterscheidet. Es hat sich gezeigt, dass sich die Verteilung von weiblichen und männlichen Teilnehmern in etwa an den Zahlen der Studienanfänger orientiert. Es ist also kein signifikanter Unterschied zwischen den Anzahlen von Teilnehmerinnen an den Vorprojekte und weiblichen Studienanfängern festzustellen.

Außerdem werden der Studiengang und die Vorkenntnisse erfasst. Um bei der Codeauswertung die Vorkenntnisse einordnen zu können, wird nach der Schulausbildung Informatik gefragt. Dieses Merkmal wird vor allem im nächsten Jahr interessant, wenn in Bayern der „Doppelte Jahrgang“ die Gymnasien verlässt. Es wird dann Schüler geben, die bereits eine schulische Grundausbildung in Informatik durchlaufen haben, und solche, denen diese Ausbildung fehlt. Hier ergeben sich interessante Forschungsfelder.

Desweiteren wird bei den Personendaten auch noch die Herkunft erfragt. Dabei gibt es die Möglichkeiten, das Bundesland anzugeben oder das Land, aus dem man kommt. Auch hier soll eine Einschätzung der informatischen Vorbildung stattfinden.

4.2 Veranstaltung

Der zweite Teil des Fragebogens beschäftigt sich mit der Veranstaltung an sich. Hier werden Merkmale wie Zufriedenheit mit Ankündigung, Organisation, Ablauf und Gruppengröße abgefragt. Diese Merkmale dienen in erster Linie der Verbesserung der Durchführung und werden deswegen hier nicht näher betrachtet.

4.3 Blätter

Im Abschnitt „Blätter“ soll vor allem die Verständlichkeit der Arbeitsblätter eingeschätzt werden.

In der ersten Frage sollen die Teilnehmer den Informationsgehalt der Arbeitsblätter auf einer Skala von 1–5 (hoch – niedrig) einstufen. 84% haben den Informationsgehalt mit 1–3 eingestuft, wobei der Hauptanteil mit 40% bei Stufe 2 liegt.

Die zweite Frage zielte auf die Verständlichkeit der Blätter. Hier ging die Skala von 1–5 (gut – schlecht). Die Verteilung der Antworten ist genau die gleiche wie bei der vorherigen Frage.

Die Ausführlichkeit der Erklärungen ist Inhalt der dritten Frage. Hier ging die Skala von 1–5 (nicht ausführlich – zu ausführlich). Auch hier haben 88% zwischen 1–3 bewertet. Der Hauptanteil liegt hier mit 61% aber auf 3 – „ausführlich“.

In der Gesamtheit sind die Blätter also durchweg positiv bewertet worden. Dies deckt sich auch mit den Ergebnissen der Codeanalyse, die in Abschnitt 5 präsentiert wird.

4.4 Tutor

Der Tutor stellt einen ganz zentralen Aspekt des Konzepts dar. Von ihm hängen sowohl der Erfolg der Teilnehmer wie auch die Selbstständigkeit der Teilnehmer ab. Zum Tutor wurden zwei Fragen gestellt.

Die erste Frage zielt darauf ab, wie hilfreich die Anwesenheit empfunden wurde. Die Skala reicht von 1–5 (sehr – gar nicht). Die Antworten waren zu 95% im Bereich 1–3, wobei 65% die Stufe 1 angaben. Die Präsenz eines Tutors ist also von den Teilnehmern als extrem wichtig eingestuft worden. Dies deckt sich auch mit unseren Beobachtungen während der Durchführung.

Dies wird auch durch die zweite Frage gestützt, die auf die Verständlichkeit der Erklärungen des Tutors abzielt. Hier reicht die Skala von 1–5 (verständlich – nicht verständlich). 92% gaben die Stufen 1–3 an, wobei der Hauptanteil von 60% bei Stufe 1 lag.

4.5 Selbsteinschätzung

Der fünfte Abschnitt ist der wohl mit Abstand interessanteste. Hier wurde nach der Selbsteinschätzung der gelernten Inhalte gefragt.

Die erste Frage ermittelt, wie viel die Teilnehmer glauben gelernt zu haben. Hier reicht die Skala von 1–5 (viel – wenig). Die Ergebnisse ballen sich mit 88% wieder in den Stufen 1–3, wobei diesmal der größte Anteil mit 57% auf Stufe 2 entfällt.

Mit der zweiten Frage sollten die Teilnehmer ihre hinzugewonnenen Javakenntnisse einschätzen. Die Skala war hier umgedreht von 1–5 (nicht vorhanden – selbstständiges Programmieren). 41% der Teilnehmer haben sich in Stufe 3 eingeordnet. Immerhin 11% trauten sich nach der Veranstaltung zu, selbstständig zu programmieren.

Die letzte Frage zielt auf das Verständnis des objektorientierten Konzepts. Hier reicht die Skala von 1–5 (verstanden – nicht verstanden). Wie auch die Codeanalyse im nächsten Abschnitt zeigen wird, wurden die Grundkonzepte der Objektorientierung von 95% der Teilnehmer verstanden. Dennoch ist diese Zahl mit Vorsicht zu genießen, da sowohl der Tutor als auch die Entwicklungsumgebung, die bereits einen großen Teil vorgegeben hat, eine gewichtige Rolle gespielt haben dürften.

5 Auswertungsmöglichkeiten des Codes

In einem ersten Schritt haben wir den Code der Teilnehmer analysiert, die ohne Vorkenntnisse in die Veranstaltung gegangen sind. Dies geschah vor allem deswegen, da hier offensichtlich eine sehr homogene Gruppe bezüglich der Vorbedingungen vorliegt. Um sicherzustellen, dass die richtigen Ergebnisse untersucht werden, haben wir die Fragebögen mit Nummern versehen. Diese Nummern sollten von den Teilnehmern als Kommentar in den Quellcode geschrieben werden. Dadurch ist die Anonymität gewährleistet und die Vorbedingungen wie Geschlecht, Herkunft und Vorkenntnisse können im Nachhinein zugeordnet werden.

Für den Erfolg der Vorprojekte gibt es zwei Kriterien. Der erste ganz wichtige Punkt, der untersucht werden muss, ist die Frage, ob das Konzept überhaupt angenommen wurde. Ein Anhaltspunkt ist der Fragebogen, der im vorherigen Abschnitt ausführlich besprochen wurde.

Ein anderer Aspekt ist die Frage, welche und wie viele Teile der Arbeitsblätter umgesetzt wurden. Zunächst sollte festgehalten werden, dass von 37 Projekten, die untersucht wur-

den, nur zwei nicht compilierbar waren. Einschränken muss man diese Aussage nur durch die Tatsache, dass die benutzten Entwicklungsumgebungen eine große Hilfe beim Erstellen von compilierbarem Code sind.

Weitaus interessanter ist die Tatsache, dass 26 von 37 Projekten eine sinnvolle Ausgabe auf vorher bestimmte Testszenerarien gaben. Dafür wurden die Programme mit zwei Eingaben getestet, einem Normalfall und einem Spezialfall. Der Spezialfall wurde dabei weit weniger berücksichtigt als der Normalfall.

Zusätzlich soll noch betrachtet werden, wie die Teilnehmer die Konzepte, die auf den Arbeitsblättern angesprochen wurden, verwendet haben.

Diese Konzepte sind im Einzelnen:

- C1) **Anordnung von Attributen, Konstruktoren und Methoden:** Auf dem zweiten Arbeitsblatt gibt es eine Beispielklasse, die die Anordnung der einzelnen Elemente demonstriert.
- C2) **Initialisierung der Attribute mit Standardwerten:** Im Text wird explizit darauf hingewiesen, dass bereits im Konstruktor die Attribute initialisiert werden sollten. Darüber hinaus sollten generell alle Attribute initialisiert werden. Eine besondere Herausforderung stellen dabei die Arrays dar.
- C3) **Definition eines eigenen Konstruktors mit Parametern:** Im Abschnitt über Konstruktoren wurde auch darauf eingegangen, dass dieser überladen werden kann.
- C4) **Verwendung eines Rückgabewertes bei Methoden:** Haben die Teilnehmer das „return“-Statement benutzt oder haben sie die Rückgabe mit globalen Variablen realisiert?
- C5) **Verwendung von Parametern im Methodenkopf:** Haben die Teilnehmer Parameter im Methodenkopf benutzt oder haben sie nur globale Variablen verwendet?
- C6) **Korrekte Verwendung von Zugriffsmodifikatoren:** Nicht alle Methoden und Attribute müssen public sein. Die Datenkapselung ist ein zentrales Element der Objektorientierung.
- C7) **Verwendung von Feldern:** Felder wurden auf dem Arbeitsblatt extra hervorgehoben.
- C8) **Verwendung von „this“:** Der Gebrauch von sprechenden Attributnamen zwingt häufig dazu, die gleichen Bezeichner für Attribute und Parameter zu benutzen.
- C9) **Verwendung der „main“-Methode:** Zu Beginn des Projektes sollten alle Studenten mit BlueJ arbeiten. Hier ist der direkte Zugriff auf die Objekte ohne eine „main“-Methode möglich. Am Ende des Projektes sollten die Teilnehmer dann ihren Code ohne BlueJ ausführen. Dazu ist die Implementierung einer „main“-Methode nötig.
- C10) **Korrekte Verwendung von Entscheidungs-Konstrukten:** Die Teilnehmer sollten das „if“-Konstrukt korrekt mit einem „else“-Zweig benutzen, sofern dieser nötig ist. (Nicht mit zwei „if“-Konstrukten)
- C11) **Verwendung von Wiederholungsschleifen:** Auf den Arbeitsblättern wurden drei verschiedene Schleifenarten vorgestellt. Diese sind „while-do“, „do-while“ und „for“.

Wie man in Abbildung 2 sehen kann, gibt es einige Konzepte, die den Teilnehmern leichter gefallen sind, und andere, die kaum verwendet wurden. So hat nur ein Teilnehmer einen eigenen Konstruktor (C3) verwendet. Ebenso wurden kaum Rückgabewerte (C4) oder Parameter (C5) in Methoden benutzt. Dies deutet darauf hin, dass der Einsatz

globaler Variablen wohl einfacher umzusetzen ist. Die globalen Variablen wurden benutzt, ohne auf den Blättern erwähnt zu werden.

Das wohl einfachste Konzept sind die Wiederholungsschleifen (C11), die von fast allen Teilnehmern verwendet wurden. Insgesamt betrachtet wurden aber doch mehr als die Hälfte der Konzepte von den meisten Teilnehmern umgesetzt.

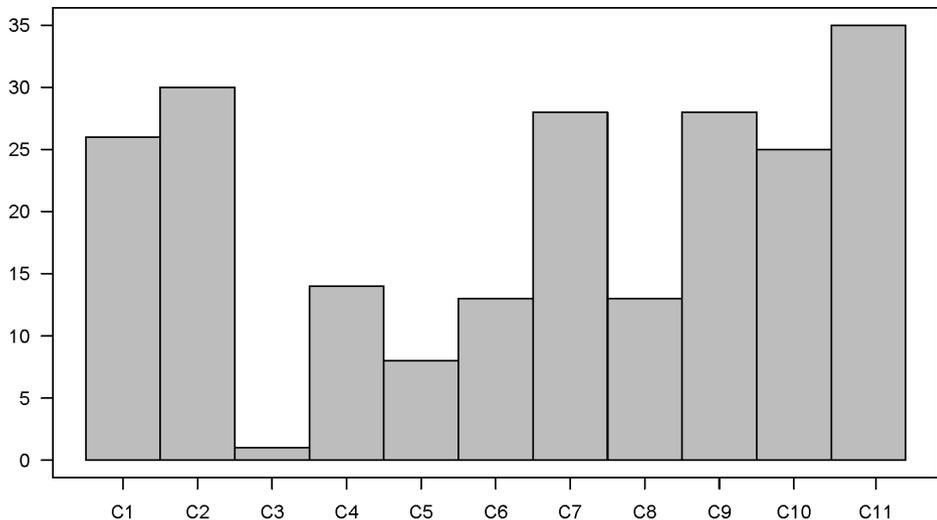


Abbildung 2: Häufigkeit der Konzepte

Der zweite wichtige Punkt ist die Auswirkung der Ergebnisse auf den Studienverlauf. Sind also die Teilnehmer ohne Vorkenntnisse im Laufe der Einführungsveranstaltung besser als diejenigen, die nicht an den Vorprojekten teilgenommen haben? Dieser Frage werden wir in den nächsten Semestern zusammen mit einer Untersuchung des „Doppelten Jahrgangs“ nachgehen.

6 Ausblick

Die allgemeine Resonanz der Teilnehmer an den Vorkursen ist durchweg positiv. Betrachtet man die Ergebnisse, die von den Teilnehmern produziert wurden, so kann man sagen, dass trotz geringer Vorkenntnisse von nahezu allen Teilnehmern ein lauffähiges Programm produziert wurde. Die Ergebnisse der ersten beiden Runden haben schon einige interessante Fragen aufgeworfen.

Im nächsten Jahr können wir die gewonnen Ergebnisse dann in einem besonders spannenden Szenario anwenden. In Bayern wird dann der sog. „Doppelte Jahrgang“ die Schulen verlassen. Hier treffen im gleichen Jahr Schüler aufeinander, die bereits mehrere Jahre Informatikunterricht gehabt hatten, wie auch solche, die noch keine oder kaum Vorkenntnisse haben [HP1]. In diesem Forschungsfeld lassen sich hoffentlich Ergebnisse erarbeiten, die auf Lernstrategien in der Informatik Rückschlüsse zulassen.

Literatur

- [BBK1] J. Bergin, K. Bruce, M. Kölling: Objects-Early Tools – A Demonstration, SIGCSE 2005, St. Louis – Missouri – USA, 2005
- [KR1] M. Kölling, J. Rosenberg: Guidelines for Teaching Object Orientation with Java, ITiCSE 2001, Canterbury – UK, 2001
- [MW1] H. Müller, F. Weichert: Vorkurs Informatik - Der Einstieg ins Informatikstudium, B.G. Teubner Verlag – Wiesbaden, 2005
- [WH1] H. M. Walker: The Role of Programming in Introductory Computing Courses, ACM Inroads Vol.1 No.2, 2010
- [DT1] I. Donchev, E. Todorova: Object-Oriented Programming in Bulgarian Universities' Informatics and Computer Science Curricula, Informatics in Education 2008, Vilnius, 2008
- [CB1] M. E. Caspersen, J. Bennedsen: Instructional Design of a Programming Course – A Learning Theoretic Approach, ICER 2007, Atlanta – Georgia – USA, 2007
- [HC1] H. B. Christensen: Implications of Perspective in Teaching Objects First and Object Design, ITiCSE 2005, Monte de Caparica – Portugal, 2005
- [HS1] J.S. Hares, J. D. Smart: Object orientation: technology, techniques, management and migration, John Wiley & Sons Ltd, Chichester, 1994
- [HP1] P. Hubwieser: Functions, Objects and States: Teaching Informatics in Secondary Schools, 2nd International Conference ISSEP, Springer, Berlin, 2006
- [SB1] K. Bender, M. Steinert: Ein handlungsorientiertes, didaktisches Training für Tutoren im Bachelorstudium der Informatik, HDI 2008 – 3. Workshop des GI-Fachbereichs Ausbildung und Beruf/Didaktik der Informatik, Universitätsverlag Potsdam, Potsdam, 2008
- [AK1] L. W. Anderson, D. R. Krathwohl: A Taxonomy for Learning, Teaching, and Assessing – A Revision of Bloom's Taxonomy of Educational Objectives, Longman, 2001
- [BLG1] N. R. Boyer, S. Langevin, A. Gaspar: Self Direction & Constructivism in Programming Education, SIGITE 2008, Cincinnati – Ohio – USA, 2008

Eine aufwandsbeschränkte Einführung in die modellbasierte Softwareentwicklung

Timo Kehrer, Udo Kelter

Praktische Informatik – Universität Siegen
57068 Siegen - Germany
Web: pi.informatik.uni-siegen.de
E-Mail: kehrer@informatik.uni-siegen.de,
kelter@informatik.uni-siegen.de

Zusammenfassung: Seit einigen Jahren gewinnt die modellgetriebene bzw. modellbasierte Entwicklung (MBSE) immer mehr an praktischer Bedeutung. Über die grundlegende Modellierungskompetenz hinaus sollte ein Informatikstudium daher auch Konzepte und Technologien der MBSE vermitteln, und zwar nicht nur vage verstandene Schlagworte, sondern die Kompetenz, eine konkrete MBSE-Infrastruktur praktisch einzusetzen. Aufgrund der Komplexität der Technologien ist es sehr schwierig, dieses Ziel mit beschränktem Aufwand zu erreichen. Dieses Papier stellt eine Unterrichtssequenz vor, mit der ein substantieller Qualifikationsgrad mit relativ geringem Aufwand (unter 30 Arbeitsstunden bzw. einem Leistungspunkt) erreicht wird.

1 Einleitung und Motivation

Um die Komplexität großer Systeme zu beherrschen, hat die Modellbildung als Mittel der Abstraktion und Problemanalyse längst einen festen Platz in der Informatik eingenommen [7, 11]. Dementsprechend definiert u.a. der „IEEE Guide to the Software Engineering Body of Knowledge“ [5] für den Themenkomplex Modellierung hohe bis sehr hohe Kompetenzstufen als Ziel einer Informatikausbildung an Hochschulen.

In klassischen Entwicklungsprozessen fungieren Modelle primär als autarkes Dokumentations- oder Spezifikationsmittel. Seit einigen Jahren gewinnt darüber hinaus die modellgetriebene bzw. modellbasierte Entwicklung (MBSE) immer mehr an praktischer Bedeutung; ohne sie sind manche komplexen Produkte kaum noch realisierbar. Modelle werden hier in mehreren, ganz oder teilweise automatisierten Schritten zu lauffähiger Software oder zu interpretierbaren Spezifikationen verfeinert. Hierdurch ergeben sich zwei neue Ausbildungsziele, die deutlich über die grundlegende Modellierungskompetenz, welche für klassische Entwicklungsprozesse vermittelt wird und die wir i. F. als Grundlage voraussetzen, hinausgehen:

1. Absolventen sollten zumindest die „einfache“ Nutzung einer MBSE-Technologie beherrschen und einen eigenen Eindruck von den Vorteilen und Grenzen der MBSE-Technologien gewinnen. Selbst ein solcher erster Einstieg stellt wegen der Komplexität der Werkzeuge eine Herausforderung dar und erfordert zumindest ein grundsätzliches Verständnis der Technologien.
2. In der Praxis müssen die MBSE-Technologien häufig an die konkreten Umstände angepasst werden; im Extremfall läuft dies auf die Entwicklung domänenspezifischer Sprachen (DSLs) hinaus. Hierzu ist ein noch deutlich tieferes Verständnis der MBSE-Technologien erforderlich.

Die Komplexität des Themas ist auch am Umfang von Lehrbüchern zum Thema MBSE (bzw. MDA) abzulesen: typisch sind 300–500 Seiten, s. Literaturdiskussion in Abschnitt 4. Dies entspricht 4–8 ECTS-Leistungspunkten, die allenfalls im Wahlpflichtbereich des Curriculums unterzubringen sind. Um das Thema im Pflichtbereich unterzubringen bzw. viele Studierende zu erreichen, wird eine sehr kompakte Einführung benötigt.

In diesem Papier stellen wir eine Unterrichtssequenz vor, welche die erste o.g. Qualifikationsstufe erreicht und nur 20–30 Stunden Arbeitsaufwand für die Lernenden, entsprechend ca. 1.0 ECTS-Punkten, verursacht. Die Unterrichtssequenz ist gedacht für Studierende im 3. oder 4. Fachsemester, die schon über grundlegende Modellierungskompetenzen verfügen und wenigstens zwei Modelltypen kennengelernt haben – neben Datenmodellen (i.d.R. Klassendiagramme der UML) zusätzlich einen weiteren (Petri-Netze, Zustandsautomaten o.ä.). Wir haben diese Sequenz sowohl als selbstständigen Kurs wie auch integriert in eine Softwaretechnik-Pflichtvorlesung angeboten.

Nach unseren bisherigen Erfahrungen treten beim Unterricht von MBSE-Technologien zwei Hauptprobleme auf: die Vielzahl voneinander abhängiger Begriffe und die semantischen (Meta-)Ebenen der auftretenden Daten, die sehr leicht verwechselt werden. Die Herausforderung an die Unterrichtssequenz liegt daher darin, die anfangs schwer überschaubaren Lerninhalte zu entwirren und schrittweise aufzubauen.

In Abschnitt 2 werden die angestrebten Qualifikationen detaillierter aufgeschlüsselt. Der Ablauf der Unterrichtssequenz wird in Abschnitt 3 ausführlich beschrieben, ferner die damit gemachten Erfahrungen. Abschnitt 4 diskutiert andere Ansätze, in das Themengebiet MBSE einzuführen. Wir schließen das Papier mit einem Fazit in Abschnitt 5.

2 Lernziele einer MBSE-Einführung

Im Folgenden werden die für das abstrakte Lernziel, MBSE prinzipiell verstanden zu haben und praktisch anwenden zu können, notwendigen Qualifikationen detailliert aufgeschlüsselt. Wir beginnen mit evtl. nachzuholenden Vorkenntnissen. Anschließend diskutieren wir die Aspekte Arbeitsumgebung, begriffliche Grundlagen und Werkzeug-Funktionen, welche im Rahmen einer kompakten MBSE-Einführung jeweils auf eine geeignete Auswahl reduziert werden müssen.

2.1 Nachzuholende Vorkenntnisse

Belastbare Kenntnisse in der Datenmodellierung setzen wir wie bereits erwähnt voraus. Diesbezügliche Schwächen wirken sich nach unserer Erfahrung äußerst negativ auf den Lernerfolg aus. Sofern diese Kenntnisse bei den Teilnehmern noch unsicher sind oder deren Ersterwerb länger zurückliegt, müssen diese Kenntnisse unbedingt vorab aufgefrischt und gefestigt werden. Geeignete Maßnahmen sind die Wiederholung früherer Übungen oder die Einführung von einigen einfachen Analysemustern, z.B. gem. [6]. Bei dem zweiten Modelltyp ist der Beherrschungsgrad weniger kritisch.

2.2 Auswahl der praktischen Arbeitsumgebung

Das Lernziel, MBSE praktisch anwenden zu können, erfordert natürlich, sich für ein konkretes MBSE-Framework zu entscheiden. Unsere Wahl fiel auf das Eclipse Modeling Framework (EMF) [2, 3, 13]. EMF zählt zu den bekanntesten MBSE-Frameworks, ist

frei verfügbar und dort, wo Eclipse als Entwicklungsumgebung eingesetzt wird, besonders naheliegend.

Auf die Lernprobleme im Zusammenhang mit MBSE hat die Wahl von EMF u.E. nur geringen Einfluss, d.h. die Lernprobleme treten bei anderen MBSE-Frameworks sehr ähnlich auf. Viele Lernprobleme liegen eher in der Komplexität der Konzepte, weniger in deren spezieller Implementierung in EMF. Die durch EMF selbst verursachten Lernprobleme liegen in der für Anfänger zu hohen Funktionsvielfalt und daraus resultierenden, unüberschaubaren Menüs sowie stellenweise irreführend, um nicht zu sagen falsch gewählten Bezeichnungen („Generatormodell“). Diese Probleme lassen sich indessen durch überschaubare Gegenmaßnahmen lösen.

2.3 Auswahl der zu erlernenden Begrifflichkeiten

Die zu erlernenden Prinzipien und Begriffe der MBSE können grob in zwei Bereiche unterteilt werden: (a) abstrakte, werkzeuginabhängige begriffliche Grundlagen: hierzu gehören die Unterscheidung von Transformation und Interpretation von Modellen, typische Transformationsketten, Repräsentation von Modellen und Metamodellierung, ferner Randthemen wie Produktlinien und generative SW-Entwicklung; (b) konkrete Ausprägungen der abstrakten Begriffe und ergänzende Details im benutzten MBSE-Framework, insb. die konkrete Funktion von Modelltransformatoren, die Nutzung von Ecore zur Modellrepräsentation und die Werkzeugbedienung.

Hier stellt sich vor allem die Frage, wie umfangreich abstrakte Begriffe überhaupt und welche davon vor den praktischen Unterrichtsanteilen eingeführt werden sollen. Auf keinen Fall sollte versucht werden, begriffliche Grundlagen in einem Umfang vorab einzuführen, der vergleichbar mit [8] ist. Dort wird die komplette Vielfalt möglicher Transformationsmethoden und -ketten beschrieben, ohne sich auf konkrete Technologien festzulegen, um die volle Bandbreite der Anwendungsmöglichkeiten aufzuzeigen. Anfängern sind die vielen, vage angedeuteten Technologien indes unbekannt, und die Vielfalt der Varianten ist zunächst verwirrend. Selbst mit einer Einführung, die nur eine stark reduzierte Menge an Begriffen enthielt, wurden in früheren Veranstaltungen schlechte Erfahrungen gemacht: Viele Teilnehmer hatten große Probleme, die vorab erklärten, abstrakteren Strukturen im EMF-Kontext wiederzuerkennen¹. Anders gesagt hat die theoretische Einführung kaum Vorteile für die spätere praktische Einführung gebracht.

Im Endeffekt plädieren wir für die Anwendung einer induktiven Unterrichtsmethode, bei der abstrakte Begriffe zuerst nur im Kontext konkreter Ausprägungen (also konkreter Werkzeuge, Metamodelle usw.) eingeführt werden. Erst wenn diese konkreten Ausprägungen relativ gut verstanden sind, kann eine begrenzte Anzahl abstrakterer Begriffe erneut behandelt und verselbständigt werden, indem alternative Ausprägungen skizziert werden.

2.4 Auswahl der zu erlernenden EMF-Funktionen

Die möglichen Einsatzgebiete für EMF sind vielfältig. Neben den beiden Grundfunktionen als Quelltextgenerierungs-Framework sowie als Modellierungs- und Datenintegrations-Framework existieren eine Reihe darauf aufbauender Funktionen. Nahezu alle Operationen auf Modellen, wie bspw. die Transformation von Modellen, der Modellver-

¹ Ferner wurden zugehörige Verständnisfragen in der Klausur meist nicht oder nicht korrekt beantwortet.

gleich, die Validierung von Modellen oder Abfragen auf Modellen, sind in Form konkreter Technologien auf Basis von EMF verfügbar. Einige dieser Technologien sind i.d.R. auch in den Arbeitsumgebungen der Studenten installiert. In Verbindung mit der ohnehin großen Funktionsvielfalt der Entwicklungsumgebung führt dies insbesondere bei unerfahrenen Studenten zu einer Reizüberflutung. Umso wichtiger ist es, sich anfangs klar auf die essentiellen Grundfunktionen zu konzentrieren. Hierzu zählen unserer Ansicht nach:

- die Erstellung von Ecore-Modellen mittels des grafischen Diagramm-Editors bzw. mittels des Baum-basierten Editors
- die Erstellung und Bearbeitung von Ecore-Modell-Instanzen mittels des reflektiven Editors
- die Generierung von Java-Quelltext
- die rudimentäre Anpassung des generierten Quelltextes durch Steueranweisungen (genmodel) für den Code-Generator
- die Benutzung der EMF API: obligatorisch sind hier das Laden und Speichern von EMF-Ressourcen sowie die Nutzung des generierten Quelltextes. Optional ist die Nutzung der reflektiven EMF API.

3 Struktur einer „minimalen“ MBSE-Einführung

Im Folgenden beschreiben wir den zeitlichen Aufbau sowie die vermittelten Lerninhalte der Unterrichtssequenz. Anschließend skizzieren wir Evaluationsmöglichkeiten und unsere bisherigen Erfahrungen.

3.1 Auswahl der zu erlernenden EMF-Funktionen

Abbildung 1 zeigt eine Übersicht über die Lektionen der Unterrichtssequenz, welche im weiteren Verlauf dieses Abschnitts näher erläutert werden. In der Spalte „Zeitbedarf“ gibt „P“ den Zeitbedarf für die Präsenzveranstaltung, „U“ den Zeitbedarf für selbstständig zu bearbeitende Übungen einer jeweiligen Lektion an.

Zeitbedarf	Kurzbeschreibung	Lernziele	
		Konzeptuell	Technisch
1 P: 90 min. U: ca. 6h	Modell zu Code-Transformation	Generierungsansatz der MBSE	Grundlegende EMF-Funktionen: (1) Erstellung von Ecore-Modellen und -Instanzen (2) Generierung von Java-Quelltext (3) Nutzung der EMF API
2 P: 180 min. U: 180 min.	Parametrisierung der Modell zu Code-Transformation	(1) Einflussnahme auf generierten Quelltext durch Steueranweisungen für den Generator (2) Einordnung der MDA-Begriffe PIM und PSM	(1) EMF Generator-Modell (genmodel) (2) Vertiefung von EMF API-Kenntnissen
3 P: 90 min. U: 180 min.	Perspektivenwechsel auf die Metamodellierungsebene	Kennenlernen des Metamodells eines konkreten Modelltyps in mehreren technologiegebundenen Varianten	-
4 P: 90 min. U: ca. 8h	Interpreteransatz der MBSE	Erkenntnis für die aus technischer Sicht identische Verarbeitung von konzeptuell auf verschiedenen Meta-Ebenen angesiedelten Daten bzw. Modellen	-
5 P: 90 min.	Reflexion & Einordnung	Einordnung der anhand von Beispielen eingeführten Konzepte in den terminologischen und konzeptuellen Rahmen der MBSE	-

Abb. 1: Lerninhalte und Ablauf: Übersicht

1. Lektion: Modell-zu-Code-Transformation im Kontext einer fachlichen Domäne

Ziele: Konzeptuell sollen die Teilnehmer in dieser Stunde anhand eines konkreten, alltäglichen Beispiels kennenlernen, wie aus einem (Daten-) Modell Quelltext generiert wird (Generierungsansatz der modellbasierten Entwicklung).

Technisch lernen die Teilnehmer die grundlegenden Funktionen von EMF kennen. Sie sollten nach dieser Stunde in der Lage sein, selbständig Ecore-Modelle und -Instanzen zu erstellen, zu editieren und daraus Java-Quelltext zu generieren. Auch der Umgang mit einer Teilmenge der EMF-API (Laden und Speichern von Modellen) sowie die Benutzung der auf Basis des Modells generierten Schnittstellen sollte nach dieser Stunde in Grundzügen beherrscht werden. Ein Einblick in die Implementierung der generierten Schnittstellen wird erst in der zweiten Unterrichtseinheit (s. unten) vorgenommen.

Methoden/Ablauf: Zu Beginn ist in einem einführenden Vortrag ein kurzer Überblick über die EMF-Modellierungstechnik Ecore sowie das Prinzip der „Modell zu Code“-Transformation in EMF zu liefern. Der Vortrag sollte dabei einen Umfang von 5–10 Folien nicht überschreiten. Die wesentlichen Konzepte können später anhand der Erstellung der konkreten Beispiel-Applikation sehr gut illustriert werden. Es empfiehlt sich dringend, eine solche Einführung im Rechnerpool vorzunehmen, so dass die benötigten Werkzeugfunktionen direkt nachvollzogen werden können. Details über die Funktionsweise des Codegenerators sind weitestgehend zu vermeiden.

Die fachliche Domäne sollte möglichst einfach sein, so z.B. das im EMF-Tutorial [14] eingeführte „Bibliothekswesen“. Das Tutorial kann um die Erstellung einer einfachen

Applikation erweitert werden, für welche die Datenhaltungsschicht mittels EMF vollständig generiert werden kann. Der Gesamtbedarf dieser Einführung variiert, je nach Einführungstempo und Gruppengröße, zwischen 90 und 180 Minuten.

Vertieft wird der Stoff durch eine selbstständig zu bearbeitende Übungsaufgabe. Es empfiehlt sich die exakte Wiederholung der während der Präsenzübungen durchgeführten Funktionen anhand einer anderen, fachlichen Domäne. Je nach Umfang der auf der Datenhaltungsschicht aufsetzenden Applikation kann hier mit einem Bearbeitungsaufwand von ca. 4–8 Stunden kalkuliert werden.

2. Lektion: Parametrisierung der Modell-zu-Code-Transformation

Ziele: Auf konzeptueller Ebene lernen die Teilnehmer in dieser Lektion beispielhaft kennen, in welchem Rahmen die generierte Software durch Steueranweisungen für den Generator unterschiedlich aussehen kann. Ferner werden auch bestimmte Muster zur Umsetzung von Entwurfsdatenmodellen in eine objektorientierte Programmiersprache anhand der detaillierten Betrachtung des generierten Quelltextes nochmals wiederholt. Die Teilnehmer sollten nach dieser Unterrichtseinheit auch in der Lage sein, einen Bezug zu den MDA-Konzepten PIM (Platform Independent Model) und PSM (Platform Dependent Model) [8] herzustellen.

Auf technischer Ebene sollen die Teilnehmer den Sinn des Generator-Modells (genmodel), welches in der MDA-Terminologie die Funktion des PSM einnimmt, verstehen und einzelne Steueranweisungen für den Code-Generator kennen.

Methoden/Ablauf: Anhand der Diskussion verschiedener Varianten zur Umsetzung von Entwurfsdatenmodellen in eine objektorientierte Programmiersprache wird zunächst die Parametrisierbarkeit des Code-Generators motiviert. Diskutiert werden kann hier einerseits auf der Implementierungsebene (z.B. die Varianten zur Umsetzung von Mehrfachvererbungen oder Assoziationsbeziehungen in Java), andererseits aber auch auf der Ebene der generierten Schnittstellen (bspw. über Signaturen bzw. die Existenz von get- und set-Methoden). Zudem können globale Parameter wie bspw. das Zielverzeichnis oder spezifische Namenspräfixe für die generierten Quellen genannt werden. Der Effekt einzelner Steueranweisungen für den Code-Generator wird direkt am in Lektion 1 eingeführten Beispiel gezeigt. Die dazu notwendigen Anpassungen des Generator-Modells sollten sofort am Rechner nachvollzogen werden.

3. Lektion: Wechsel auf die Metamodellierungsebene

Ziele: Konzeptuell soll hier von der Modellierung einer alltäglichen Anwendung auf die Modellierung von Modellen eines gegebenen Modelltyps übergegangen werden. Dabei sollen Lösungen typischer Modellierungsprobleme wie die Modellierung von attributierten Kanten, mehrstelligen Beziehungen usw. erlernt werden. Ferner sollen die Teilnehmer technologiegebundene Repräsentationen der Modelle kennenlernen. In dieser Unterrichtseinheit werden keine neuen EMF-Funktionen erlernt.

Auswahl des Beispiel-Modelltyps: Klassendiagramme erscheinen als Anwendungsbeispiel naheliegend, sind aber völlig ungeeignet und sollten auf keinen Fall verwendet werden. Bei Klassendiagrammen kommt es sehr leicht zu Verwechslungen zwischen Modellierungsgegenstand und der zur Modellierung der (technischen) Repräsentation von Klassendiagrammen verwendeten Notation (in diesem Fall EMF-Ecore), welche den

Klassendiagrammen sehr ähnlich ist². Verhaltensmodelle sind hier deutlich geeigneter und führen zu weniger Missverständnissen. Wir empfehlen hier als Modelltyp eine einfache Variante von Zustandsautomaten und setzen dies i. f. Voraus.

Methoden/Ablauf: Diese Lektion besteht aus einer auf drei Stunden Bearbeitungszeit angelegten Übung, in der das Metamodell eines bestimmten Modelltyps zunächst konzeptuell zu entwerfen und anschließend mittels EMF-Ecore konkret umzusetzen ist. Metamodelle sollten als abstraktes Konzept hier noch nicht thematisiert werden, sondern lediglich als Datenmodelle eingeführt werden, deren modellierte Anwendungsdomäne eine Modellierungssprache bzw. -technik ist. Der Begriff des Meta-Metamodells sollte hier noch komplett vermieden werden.

Die Übung sollte in folgenden drei Schritten ablaufen:

- Begonnen wird im Rahmen der Präsenzübungen mit der Konzeption des Metamodells für Zustandsautomaten in Form eines Analyseklassendiagramms. Dies kann bspw. in gemeinsamer Diskussion an der Tafel entstehen. Das (technologiefreie) Analyseklassendiagramm ist anschließend in Eigenarbeit in die beiden nachfolgend beschriebenen technologiegebundenen Metamodelle umzusetzen:
- Erstens mittels Standardmethoden in ein relationales Schema. Diese Standardmethoden sind ggf. an dieser Stelle kurz aufzufrischen.
- Zweitens in ein Ecore-Modell; das Ecore-Modell soll direkt im Werkzeug editiert werden (z.B. mit dem grafischen Ecore-Diagrammeditor).

4. Lektion: Einführung des Interpreteransatzes der modellbasierten Entwicklung

Ziele: Ziel dieser Lektion ist die Konstruktion einer generischen Applikation am Beispiel eines Interpreters für Zustandsautomaten. Konzeptuell lernen die Teilnehmer dabei den Interpreteransatz der modellbasierten Entwicklung kennen. Die aus technischer Sicht identische Verarbeitung von konzeptuell auf verschiedenen Meta-Ebenen angesiedelten Daten bzw. Modellen sollte den Teilnehmern aufgrund der Einführung von Metamodelle als spezielle Datenmodelle nicht befremdlich erscheinen. Dies ist eine zentrale Erkenntnis dieser Lektion.

Auf technischer Ebene wird in dieser Lektion kein neues Wissen vermittelt.

Methoden/Ablauf: Als Ausgangspunkt dieser Lektion fungiert das in der vorhergehenden Lektion erstellte Metamodell für Zustandsautomaten. Um das Verständnis für das Metamodell nochmals zu überprüfen und zu festigen, können während der Präsenzübungen bereits Testdaten für den später zu implementierenden Interpreter gemeinsam erstellt werden. Ausgehend von in der gewohnten, grafischen Notation skizzierten Zustandsautomaten (z.B. an der Tafel) wird deren statische Struktur extrahiert und mittels des reflektiven EMF-Editors editiert. Bei dieser Gelegenheit können die Begriffe der abstrakten bzw. konkreten Syntax einer Modellierungssprache eingeführt werden.

Abschließend kann die Grobarchitektur des Interpreters kurz diskutiert werden. Die Implementierung sollte den Teilnehmern überlassen werden, wobei ein infrastruktureller Anteil bereits vorgegeben und während der Präsenzübungen auch vorgestellt werden sollte. Ferner wird für die Bearbeitung der Übungsaufgabe eine einfache Simulationsumgebung bereitgestellt, welche zum Testen des Interpreters verwendet werden kann.

² Dies wird insbesondere dann sehr deutlich, wenn zur Notation von Ecore-Modellen der grafische Diagramm-Editor verwendet wird.

Abb. 2 (links) zeigt die grafische Benutzeroberfläche (GUI) der Simulationsumgebung, welche nach dem Einlesen des zu simulierenden Zustandsautomaten, hier der in Abb. 2 (rechts) dargestellten Garagentorsteuerung, erscheint. Die GUI weist für jeden Trigger, der in dem Zustandsautomaten vorkommt, eine Schaltfläche auf. Ferner existieren Protokollausgaben, welche die eingetretenen Trigger, den daraufhin eingenommenen Zustand (bzw. die eingenommenen Zustände im Falle paralleler Regionen) und die auszuführenden Aktionen protokollieren. Durch manuelles Aktivieren der Trigger-Schaltflächen kann der Interpretier so schrittweise getestet werden.

Für den während der Präsenzübungen einzuführenden Anteil ist eine Dauer von 90 Minuten ausreichend. Für die Implementierung des Interpreters werden nach unseren bisherigen Erfahrungen durchschnittlich 8–10 Stunden benötigt.

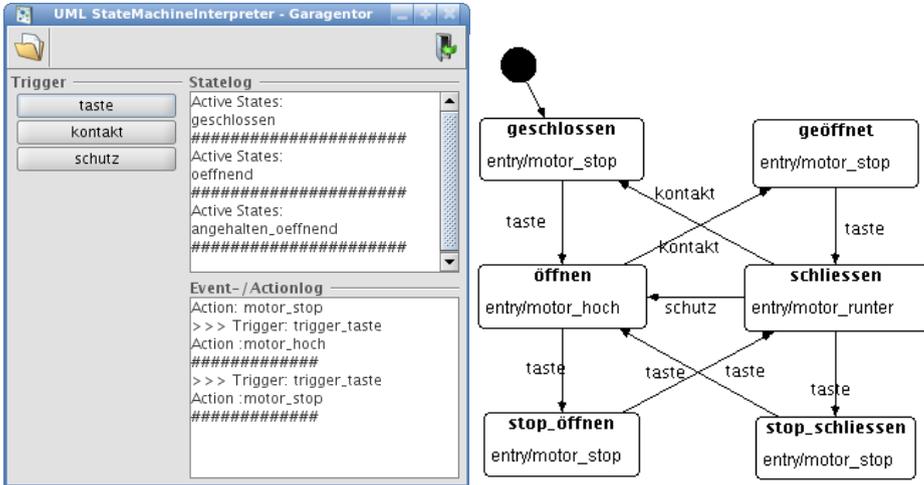


Abb. 2: GUI der Simulationsumgebung

5. Lektion: Reflexion und Einordnung

Ziele: Ziel dieser Unterrichtseinheit ist es, die während des bisherigen Verlaufs des Kurses größtenteils anhand von Beispielen eingeführten Konzepte in den terminologischen und konzeptuellen Rahmen der Modellierung und modellbasierten Entwicklung einzuordnen.

Methoden/Ablauf: Für die Reflexion des anhand von Beispielen eingeführten Stoffes empfiehlt sich ein Folienvortrag im Umfang von 45 bzw. 90 Minuten. Punktuell können dabei einige Aspekte auch erweitert werden. Insbesondere der Themenkomplex „Metamodelle“ kann hier nochmals vertieft betrachtet und in den Kontext der von der OMG spezifizierten (Meta-)Modellierungsebenen M0–M3 gesetzt werden. Optional kann hier ein kurzer Blick auf die Strukturen der UML-Spezifikationen geworfen werden. Weiterführende Themengebiete wie die Integrität von Modellen, spezialisierte Systeme für Modelltransformationen oder auch die Versionierung und Evolution von Modellen sollten kurz umrandet und in Aussicht gestellt werden.

3.2 Erste Erfahrungen

Eine auf alle Lektionen anwendbare Messtechnik zur Erfassung des Lernfortschritts ist die verbale Erfassung, entweder im Rahmen der studentischen Mitarbeit während der Präsenzphasen oder während der Abnahme bzw. Kontrolle der selbstständig zu bearbeitenden Übungsaufgaben(-Anteile). Die Archivierung der eingereichten Lösungen zu den Übungsaufgaben sowie über mehrere Semester hinweg gesammelte Klausurergebnisse bieten zudem die Möglichkeit der empirischen Validierung des Lernerfolgs.

Die bislang gesammelten, subjektiven Eindrücke sowie die qualitativen Aussagen der Teilnehmer waren durchweg positiv. Schnelle Erfolgserlebnisse bei der Arbeit mit EMF wirkten sich auf die Motivation der Teilnehmer positiv aus und abstrakte Konzepte oder Begrifflichkeiten wurden im Kern schnell verstanden. Die bewusst intensiv geführten Gespräche mit den Teilnehmern unterstreichen zudem die These, dass das bisherige Vorgehen, eine umfangreiche, theoretische Einführung vor der praktischen Umsetzung, die Teilnehmer meist überforderte.

4 Andere Ansätze einer MBSE-Einführung

Das Thema MBSE gehört zweifellos zu den fortgeschrittenen Themen der Informatik bzw. Softwaretechnik. In grundständigen, breit angelegten Softwaretechnik-Lehrbüchern wird das Thema daher zwar erwähnt, aber nur auf wenigen Seiten einführend abgehandelt. Vermittelt werden nur abstrakte Begriffe und Beschreibungen von MBSE-Infrastrukturen, nicht deren praktische Anwendung. Ähnliches gilt für breit angelegte Bücher zum Thema Architektur, z.B. [10], in denen MBSE bzw. MDA nur als einer von sehr vielen Architekturstilen abgehandelt wird, und für Bücher über die UML, die sich vor allem mit der Vielfalt an Diagrammtypen beschäftigen. Konkrete Technologien werden nicht vorgestellt, so dass Leser nicht in die Lage versetzt werden, die relativ abstrakt vorgestellten Methoden praktisch nachzuvollziehen und zu üben. Die dadurch ausbleibenden Erfolgserlebnisse führen unseren Erfahrungen nach schnell zu Frustration und gemindertem Interesse an der Thematik.

Beispiele für dedizierte Lehrbücher zum Thema MBSE, in denen praktische Kompetenzen angestrebt werden, sind [12] und [4].

Adressierte Zielgruppen von [12] sind laut eigener Aussage des Buches Architekten, Entwickler und Projektleiter. Primär verfolgtes Ziel ist es „den Leser davon zu überzeugen, dass MDSD³ ein praktikabler Ansatz ist und in der Tat in vielen Fällen der traditionellen Entwicklung überlegen ist“. Zwar wird versucht, den Leser anhand eines ausführlichen Beispiels in die Thematik einzuführen, die Einstiegshürde ist für Einführungsveranstaltungen zur Thematik MBSE allerdings zu hoch. Es werden Konzepte, Technologien und Frameworks wie bspw. Metaprogrammierung, Template-basierte Codegeneratoren oder die Java Plattform „Enterprise Edition“ (JEE) vorausgesetzt, welche von einem Studenten im 3. oder 4. Fachsemester i. d. R nicht beherrscht werden. Konkrete Technologien werden nicht vorgestellt, so dass man nicht in die Lage versetzt wird, die relativ abstrakt vorgestellten Methoden praktisch nachzuvollziehen und zu üben.

In [4] wird versucht, einen stärkeren Bezug zu konkreten Technologien basierend auf Eclipse und EMF herzustellen. Im Gegensatz zum Aufbau der von uns vorgeschlagenen Unterrichtssequenz wird hier jedoch ein deduktiver Ansatz zur Einführung in die Thema-

3 Anm.: Die Autoren verwenden den Begriff MDSD (Model Driven Software Development) synonym zu dem von uns hier verwendeten Begriff MBSE (Modellbasierte Software-Entwicklung)

tik gewählt. Die Konzeption des Buchaufbaus folgt einer Dreiteilung: Teil I beschreibt die theoretischen Grundlagen des Ansatzes, Teil II beleuchtet die Auswirkungen der vorgestellten Konzepte auf die Prozesse und die Organisation von Unternehmen. Erst Teil III bringt die Ideen zur Anwendung und setzt sie mit konkreten Technologien zur Umsetzung in Beziehung. Studenten im 3. oder 4. Fachsemester sind nach der Vermittlung von Teil I des Buches nicht in der Lage, die essentiellen Konzepte der MBSE herauszufaktorisieren. Teil II des Buches ist allenfalls in einem Studiengang Wirtschaftsinformatik nutzbar. Die vorgeschlagene a-posteriori-Illustration der theoretischen Konzepte anhand konkreter Technologien (Teil III) ist unserer Erfahrung nach meist erfolglos. Abschließend ist ferner der Anspruch des Buches zu hinterfragen, sich gleichermaßen an „Manager, Berater und Projektleiter“ wie an „Entwickler und Architekten“ als Zielgruppe zu richten.

Einen anderen Ansatz, in die Thematik einzuführen, stellen technologiegebundene Tutorials wie bspw. [14] oder [9] dar. Anfängern wird hier auf wenigen Seiten ein schneller Einstieg in komplexe Technologien ermöglicht, so dass sich schnell erste Erfolgserlebnisse einstellen. So kann das Interesse geweckt werden und es wird eher das Angebot weiterführender Tutorials oder Literatur wahrgenommen, um Kenntnisse zu vertiefen. Nachteilig ist hier jedoch zu vermerken, dass durch die Fokussierung auf konkrete Technologien eine saubere Begriffsbildung oftmals vernachlässigt bzw. Begrifflichkeiten oder Konzepte falsch oder inkonsistent eingeführt werden. Als Beispiel ist hier bereits der Titel von [9] aufzuführen, welcher mit „Create UML models and generate code“ schlichtweg falsch gewählt ist. Auch wenn die grafische Notation für Ecore-Modelle an jene für UML-Klassendiagramme erinnert, wird hier nicht aus UML-Modellen, sondern aus Ecore-Modellen Java-Quelltext generiert. Eine Einführung mittels Tutorials sollte daher nicht ohne deren Reflexion im Rahmen von Präsenzveranstaltungen stattfinden.

5 Fazit

Die hier vorgestellte Unterrichtssequenz zeigt, dass man auch mit wenig Aufwand den Studierenden zentrale Konzepte der MBSE verbunden mit praktischen Erfahrungen vermitteln kann. Insbesondere legen unsere bisher gesammelten Lehrerfahrungen auf dem Gebiet der modellbasierten Entwicklung nahe, hier eine induktive Unterrichtsmethode anzuwenden, bei der abstraktere Begriffe zuerst nur im Kontext konkreter Ausprägungen (also konkreter Werkzeuge, Metamodelle usw.) eingeführt werden. Erst wenn diese konkreten Ausprägungen relativ gut verstanden sind, kann eine begrenzte Anzahl abstrakterer Begriffe erneut behandelt und verselbständigt werden, indem alternative Ausprägungen skizziert werden.

Insgesamt lässt sich so das komplexe Thema MBSE auch schon im dritten oder vierten Fachsemester vermitteln. So kann diese zunehmend an Bedeutung gewinnende Thematik auch in Informatik-Bachelorstudiengänge integriert werden.

In Zukunft werden wir die hier beschriebene Unterrichtssequenz im Rahmen unserer grundständigen Softwaretechnik-Vorlesung, welche für die meisten Studiengänge eine Pflichtveranstaltung im 2. Studienjahr darstellt, weiter einsetzen. Durch eine gezielte und längerfristig angelegte Evaluation werden wir Aufbau und Inhalt der MBSE-Einführung weiter verfeinern mit dem Ziel, optimalen Lernerfolg bei möglichst minimalem Arbeitsaufwand der Studierenden zu erreichen.

Literatur

- [1] Eclipse UML2: Model Development Tools, UML2; www.eclipse.org/uml2/
- [2] EMF: Eclipse Modeling Framework; <http://www.eclipse.org/emf>
- [3] Gronback, Richard C.: Eclipse Modeling Project - A Domain-Specific Language (DSL) Toolkit; Addison-Wesley Longman; 2009
- [4] Gruhn, Volker; Pieper Daniel; Röttgers Carsten: MDA - Effektives Softwareengineering mit UML2 und Eclipse; Springer; 2006
- [5] IEEE Guide to the Software Engineering Body of Knowledge (SWEBOK); IEEE; 2004; <http://www.computer.org/portal/web/swebok/html/contents>
- [6] Kelter, Udo: Lehrmodul Analysemuster; Fachgruppe Praktische Informatik, Universität Siegen; 2009
- [7] Ludewig, Jochen: Models in Software Engineering - An Introduction; Softw. Syst. Model., Springer, 2:1, p.5-14; 2003
- [8] MDA Guide Version 1.0.1; OMG, Doc. [omg/2003-06-01/](http://www.omg.org/doc/2003-06-01/); 2003
- [9] Powell, Adrian: Model with the Eclipse Modeling Framework, Part 1: Create UML models and generate code; 2004; <https://www.ibm.com/developerworks/opensource/library/os-ecemf/>
- [10] Reussner, Ralf; Hasselbring, Wilhelm: Handbuch der Software-Architektur; dpunkt-Verlag, 555p.; 2009; <http://www.handbuch-softwarearchitektur.de>
- [11] Seidewitz, Ed: What Models Mean; IEEE Software 20:5, Sep./Oct. 2003, p.26-32; 2003
- [12] Stahl, Thomas; Völter, Markus; Efftinge, Sven; Haase, Arno Modellgetriebene Softwareentwicklung - Techniken, Engineering, Management; dPunkt; 2007
- [13] Steinberg, D.; Budinsky, F.; Patenostro, M.; Merks, E.: EMF: Eclipse Modeling Framework, 2nd Edition. Addison Wesley; 2008
- [14] Tutorial aus der EMF-Hilfe: Generating an EMF Model; 2007; <http://help.eclipse.org/ganymede/index.jsp?topic=/org.eclipse.emf.doc/tutorials/clibmod/clibmod.html>
- [15] UML: Unified Modeling Language; <http://www.uml.org>

Output statt Input – Zur Kompetenzformulierung in der Hochschullehre Informatik

Ralf Romeike

Pädagogische Hochschule Schwäbisch Gmünd
Oberbettringer Str. 200
73525 Schwäbisch Gmünd
E-Mail: romeike@cs.uni-potsdam.de

Zusammenfassung: Die in der Fachdidaktik Informatik im Zusammenhang mit den Bildungsstandards seit Jahren diskutierte Outputorientierung wird mittelfristig auch für die Hochschullehre verbindlich. Diese Änderung kann als Chance aufgefasst werden, aktuellen Problemen der Informatiklehre gezielt entgegenzuwirken. Basierend auf der Theorie des *Constructive Alignment* wird vorgeschlagen, im Zusammenhang mit der Outputorientierung eine Abstimmung von intendierter Kompetenz, Lernaktivität und Prüfung vorzunehmen. Zusätzlich profitieren Lehramtsstudenten von den im eigenen Lernprozess erworbenen Erfahrungen im Umgang mit Kompetenzen: wie diese formuliert, erarbeitet und geprüft werden. Anforderungen an die Formulierung von Kompetenzen werden untersucht, mit Beispielen belegt und Möglichkeiten zur Klassifizierung angeregt. Ein Austausch in den Fachbereichen und Fachdidaktiken über die individuell festgelegten Kompetenzen wird vorgeschlagen, um die hochschuldidaktische Diskussion zu bereichern.

1 Einleitung

Die Untersuchung von Konzepten und Methoden einer kompetenzorientierten Informatikausbildung stellt einen wichtigen Schwerpunkt der hochschuldidaktischen Forschung der Informatik dar. Claus [C106] stellte auf der 2. HDI-Tagung in München fest: „Eile tut Not, will man die Umstellung auf die Bachelorstudiengänge nutzen.“ Tatsächlich wurde bisher wenig darüber berichtet, wie in der deutschen Informatik-Hochschullehre die Bachelor-Umstellung genutzt wurde. Mitunter scheint der Bachelor als „verspätete Zwischenprüfung“ oder „Altersbescheinigung“ aufgefasst zu werden, in den die bisherigen Lehrveranstaltungen „irgendwie eingepasst“ werden. An verschiedenen Stellen wird darüber berichtet, wie die (verordnete) Umstellung verwirklicht wurde, selten diese aber als Chance dargestellt. Eine Outputorientierung wird noch kaum umgesetzt. So berichten [WW08] und [WDW06] zwar über konkrete Ergebnisse der Kompetenzentwicklung in ihren Lehrveranstaltungen, die Darstellung der Kompetenzen geschieht aber nach wie vor nur recht allgemein.

Von politischer Seite wird mit Einführung des Europäischen Qualifikationsrahmens [EK08] bzw. seines deutschen Pendant, dem Deutschen Qualifikationsrahmen [ADQ09] mittelfristig die Lehre so beeinflusst werden, dass die Kompetenzen, die in einem Bildungsabschnitt zu erwerben sind, konkret formuliert werden müssen. Diese Situation kann nun genutzt werden, aus hochschuldidaktischer Sicht die Chancen der Outputorien-

tierung für eine bessere Lehre auszuloten und wahrzunehmen. Eine Chance bietet sich, da mit der Outputorientierung auch ein neuer Blick auf die Lehre erforderlich wird. Die Theorie des *Constructive Alignment* ([Bi03], vgl. Kap. 3) stellt eine passende theoretische Basis für eine erfolgreiche Umstellung der Lehrveranstaltungen zur Verfügung und wird in diesem Artikel unter der Perspektive der Anwendung in der Hochschullehre Informatik dargestellt. Insbesondere wird illustriert, wie die Lehrerbildung Informatik im Besonderen von der Ausrichtung der Lehre nach dem Prinzip des *Constructive Alignment* profitieren kann. Zum anderen können die Erfahrungen und Erkenntnisse der Fachdidaktik Informatik für die Schule herangezogen werden, in der die Formulierung von Kompetenzen schon länger diskutiert wird. Abschließend wird analysiert, welche Anforderungen an die Formulierung von Kompetenzen zu stellen sind, Beispiele aus der Informatik bzgl. dieser Anforderungen untersucht und eine Orientierungsmöglichkeit vorgestellt, die formulierten Kompetenzen in einer Taxonomie zu bewerten. Damit wird eine Anregung und Diskussionsgrundlage gelegt, das Projekt „Output“ an den Hochschulen aufzugreifen.

2 Kompetenzentwicklung durch Hervorhebung des Outputs

2.1 In der schuldidaktischen Diskussion

Die Ergebnisse der PISA-Studie waren wohl der zentrale Auslöser für eine Debatte um den Output in der Schulbildung. Wurde es verpasst, die Kompetenzen der Schüler zu entwickeln, weil Lehrpläne vor allem vorgaben, welche Inhalte den Unterricht bestimmen sollten? Als Ergebnis der Diskussion sollte nun durch eine Verschiebung des Fokus weg von der Steuerung der Unterrichtsinhalte und hin zur Betrachtung der im Unterricht erworbenen Kompetenzen sichergestellt werden, dass die beabsichtigten Bildungsziele auch erreicht werden. Als Kompetenz werden „alle die bei Individuen verfügbaren oder bei ihnen erlernbaren Fähig- und Fertigkeiten, bestimmte Probleme zu lösen, [...]“ bezeichnet [We02]. Ziel der Umorientierung ist ein „standardorientiertes Unterrichten“:

Jede einzelne Unterrichtsstunde und jede Unterrichtseinheit muss sich daran messen lassen, inwieweit sie zur Förderung und Weiterentwicklung inhaltsbezogener und allgemeiner Schüler-Kompetenz beiträgt, und der Unterricht über längere Zeiträume hinweg muss so konzipiert sein, dass der Aufbau von Kompetenzen im Zentrum steht. [B106]

Zur Umsetzung dieses Ziels im Informatikunterricht wurden durch die GI-Empfehlungen Bildungsstandards verabschiedet, in welchen konkrete im Informatikunterricht zu vermittelnde Kompetenzen formuliert sind. Informatiklehrer müssen mit den Kompetenzen umgehen und diese umsetzen, allerdings auch selbst Kompetenzen als Unterrichtsziel formulieren können. Die Umstellung von inputorientierter Formulierung von Lernzielen hin zur Darstellung der intendierten Kompetenzen als Output wird damit unterstützt, auch wenn zu erwarten ist, dass die Durchsetzung in der Praxis noch einige Zeit benötigen wird.

2.2 In der hochschuldidaktischen Diskussion

Können sich auch jede Vorlesung und jedes Seminar daran messen lassen, inwieweit sie zur Förderung und Weiterentwicklung der Kompetenzen der Studenten beitragen? Im Gegensatz zur Diskussion um die Kompetenzentwicklung an Schulen scheint das traditionelle, am Humboldtschen Bildungsideal orientierte deutsche Hochschulsystem sich noch wenig mit dem *Output* zu beschäftigen. So gliedern sich Lehrveranstaltungen der

Informatik vor allem anhand ihrer Inhalte, oftmals unter besonderer Berücksichtigung der (Forschungs-)Interessen der jeweils lesenden Dozenten, aber mit nur geringer Transparenz dessen, welche Kompetenzen von den Studierenden denn tatsächlich erwartet werden. Dargestellt wird stattdessen in der Regel, welche Inhalte die Studierenden „beherrschen“ sollen. So fällt auf, dass z. B. nur wenige fachdidaktische Diskussionsergebnisse darüber vorliegen, welche Kompetenzen ein Informatikstudent nach dem Grundstudium besitzen sollte bzw. welche Kompetenzen ein Student des Lehramts Informatik zum Abschluss seines Studiums benötigt. Auch die Formulierungen vieler Modulbeschreibungen erfüllen die Anforderungen an Kompetenzformulierungen nicht.

Mit der Umsetzung der Vorgaben des Bologna-Prozesses wird nun auch für die Hochschulen eine Outputorientierung verbindlich, ermöglicht doch erst die Darstellung der tatsächlich nachgewiesenen Kompetenzen die Vergleichbarkeit und Übertragung und Anrechenbarkeit von Studienleistungen zwischen verschiedenen Universitäten. Die Umstellung stellt sich allerdings als längerfristige Aufgabe dar: „It takes a considerable change in faculty culture to adopt a language of learning outcomes, one reason that the desired convergences of Bologna will continue well beyond 2010.“ [Ve05]

Die Modularisierung der Studiengänge und die damit einhergehende Straffung des Studiums verlangt von den Studierenden ein zielstrebiges Studium. Regelmäßig ist aber nicht ersichtlich, was konkret von den Studierenden erwartet wird. Deutlich wird fehlende Klarheit über das Informatikstudium an der hohen Zahl der Studienabbrecher (vgl. [RS06]): Anforderungen der Lehrveranstaltungen sind oftmals zu Beginn nicht überschaubar, was die Einschätzung dessen, was der einzelne Student bewältigen kann, zusätzlich erschwert. Ebenso wird regelmäßig von hohen Durchfallquoten in Grundstudiumslehrveranstaltungen der Informatik berichtet. Claus [Cl06] bezeichnet dies als „Schnittstellenkomponente“: Hochschuldidaktik der Informatik hat „die speziellen Probleme beim Übergang einer ‚konsumierenden Schulzeit‘ zu einem ‚selbst aktiven und eigenverantwortlichen, fordernden Studium‘ mitzubehandeln.“ Mit Hilfe von Studieneingangstests wird versucht, Studierenden eine bessere Einschätzung ihres möglichen Studienerfolgs in der Informatik zu ermöglichen; mehr Transparenz der im Studium zu erwerbenden Kompetenzen und deren Verflechtung wären eine weitere Maßnahme zur Verbesserung der Orientierung im Informatikstudium.

Neben dem Problem des Übergangs muss sich die Hochschullehre mit der veränderten Motivation zum Studium auseinandersetzen: „Die ‚Vermassung‘ des Bachelorstudiums wird in den kommenden Jahrzehnten dazu führen, dass sich der Bachelor (aus heutiger Sicht) als verlängertes Abitur mit zusätzlichen Fachkenntnissen darstellen wird.“ [Cl06]. Biggs [Bi03] illustriert das Problem anhand zweier prototypischer Grundstudiumsstudenten. *Susan* sei eine typische „gute“ (intrinsisch motivierte) Studentin: intelligent, immer gut vorbereitet, zielorientiert, interessiert und motiviert, ihre Denk-, Schreib- und Lernfertigkeiten sind gut entwickelt. *Robert* dagegen hat deutlich weniger akademisches Interesse; ebenso fehlen ihm viele Fertigkeiten, die allgemein unter Studierfähigkeit subsumiert werden. Sein Ziel ist es, mit so wenig Aufwand wie möglich durch das Studium und damit zu einem (möglichst gut bezahlten und sicheren) Arbeitsplatz zu kommen; Robert ist extrinsisch motiviert. Brabrand und Andersen [BA06] stellen fest, dass aufgrund der gestiegenen Anzahl der Studienanfänger der Anteil der *Susans* an den Universitäten deutlich abgenommen hat. Entsprachen 1980 noch drei Viertel der Studienanfänger dem *Susan*-Typus, sind diese heute deutlich in der Minderheit – gegenüber einer Mehrzahl von Studienanfängern, deren Studierfähigkeit und -interesse als eher unzureichend zu beurteilen sind. Verschiedene Universitäten versuchen der mangelnden Studierfähigkeit mit Hilfe von Vorbereitungskursen und Mentorenprogrammen entgegenzusteu-

ern. Betrachtet man die Motivation der *Roberts*, wird deutlich, dass solche Studenten damit vermutlich nicht erreicht werden.

Für die Informatiklehre stellen sich Orientierungs- und Motivationschwierigkeiten als größte Herausforderung für das Grundstudium dar. Eine Neuausrichtung der Lehre im Zuge der Umstellung zur Outputorientierung kann diesen Problemen entgegenen.

2.3 Outputorientierung im Lehramtsstudium

Für zukünftige Lehrer ist die inhaltsbezogene Darstellung der Studienanforderungen besonders problematisch, sollen sie doch selbst Kompetenzen outputorientiert formulieren. Hier wird eine Chance verpasst, anhand der eigenen Lernerfahrung Sinn und Verständnis für die Outputorientierung zu vermitteln. Brabrand und Dahl [BD09] verdeutlichen das Problem prägnant:

Course descriptions are often lists of topic areas the students are to “learn about”, but is “to learn (to do)” the same as “to learn about”? Take for instance cooking. To “learn to cook” is rather different than to “learn about cooking”.

Die Erfahrungen der Schulpraktischen Studien und Didaktik-Lehrveranstaltungen zeigen, dass viele Studenten bei der Entscheidung dieser Frage für ihren eigenen Unterricht Defizite besitzen. Bei der Formulierung der intendierten Kompetenzen wird Outputorientierung regelmäßig als „anders formulierte Lernziele“ missverstanden.

Es ist zu erwarten, dass der Umgang mit der Outputorientierung deutlich erleichtert würde, wenn die Lehramtsstudenten bereits im Studium mit den entsprechenden Formulierungen in eigenen Lernprozessen Erfahrungen machen würden.

Für die Schule stellen Fachdidaktik und Informatiklehrer [KS09] hinsichtlich der in der Kompetenzformulierung zu verwendenden Operatoren¹ fest: *Die beim Formulieren der Aufgaben verwendeten Operatoren müssen im Unterricht eingeführt und ihr Gebrauch an verschiedenen Beispielen geübt sein. Durch die Benutzung der Operatoren soll den Schülerinnen und Schülern klar werden, welche Tätigkeiten und welche Lösungsdarstellung von ihnen erwartet werden. Mit dem konsequenten Einsatz der Operatoren wird Missdeutungen von Aufgabenstellungen entgegengewirkt.*

Es bleibt auszuloten, inwiefern das im Studium forciert werden kann und sollte.

Erfahrungen mit der Umstellung der Curricula, insbesondere der Lehre in der Informatik und den Naturwissenschaften, dokumentiert Brabrand [Br08] für die Universität Kopenhagen. Zugrunde gelegt wurde die Theorie des *Constructive Alignment* ([Bi03], vgl. Kap. 3); es wurden Kompetenzen formuliert und für die Studenten transparent gemacht. Festgestellt wird, dass die Neuausrichtung des Kurses sowohl dem Dozenten ermöglichte, seine Methoden fundiert zu reflektieren und zu verbessern und der Kurs gleichzeitig sehr gutes Feedback durch die teilnehmenden Studierenden erhielt.

Mit Hilfe der vorgenommenen Konkretisierung der Kompetenzen und der Einordnung dieser in die SOLO-Taxonomie (vgl. Kap. 6) konnten Brabrand und Dahl [BD07; BD09] zusätzlich die Entwicklung der Kompetenzstufen im Studium analysieren und Kompetenzcharakteristika der verschiedenen Fächer vergleichen.

¹ „Ein Operator ist ein Aufforderungsverb wie z. B. erläutern, darstellen oder begründen, dessen Bedeutung im Fachkontext möglichst genau spezifiziert wird.“ [KS09]

3 Constructive Alignment

Grundlegend für eine erfolgreiche Lehre ist ein Verständnis dafür, wie Studenten lernen [Ni10]. Aufbauend auf den Kerngedanken des Konstruktivismus, nach welchem Lernen als aktiver Prozess verstanden wird, hat Biggs' Theorie des *Constructive Alignment* [Bi03] das primäre Ziel, die Intention des Dozenten, die Aktivitäten der Studenten und die Prüfung so aufeinander abzustimmen, dass die Intention des Lehrenden erreicht wird. Ein als „Back-wash-effekt“ bezeichnetes Problem in nicht abgestimmten Lehrveranstaltungen ist es, dass Studenten vor allem das tun und das lernen, was in der Prüfung von ihnen verlangt wird (vgl. Abb. 2). Sollen die Studenten bspw. lernen, Sachverhalte zu analysieren und zu vergleichen, in der Prüfung wird aber nur das Wiedergeben und Identifizieren von Merkmalen gefordert, wird sich hierauf auch die Lernaktivität beschränken. Durch die Abstimmung der drei Komponenten erhalten die *Roberts* den Anreiz, sich die tatsächlich intendierten Kompetenzen anzueignen. Unterstützung gibt die Ausrichtung von Lehre und Kompetenz: In der Lehrveranstaltung müssen Analysieren und Vergleichen als Teil der Lernaktivität der Studenten berücksichtigt werden.² Hierzu muss in der Lehrveranstaltung Raum eingeräumt werden, z.B. indem die Studierenden aufgefordert werden, in der Lehrveranstaltung einen Sachverhalt in Kleingruppen zu diskutieren.

Ein Kurs erfüllt dann die Anforderungen des *Constructive Alignment*, wenn

- die intendierten Kompetenzen (IK)³ deutlich angegeben sind,
- die IK explizit den Studenten mitgeteilt werden,
- die Leistungsüberprüfung den IK entspricht,
- die Ausgestaltung der Lehre den IK entspricht.

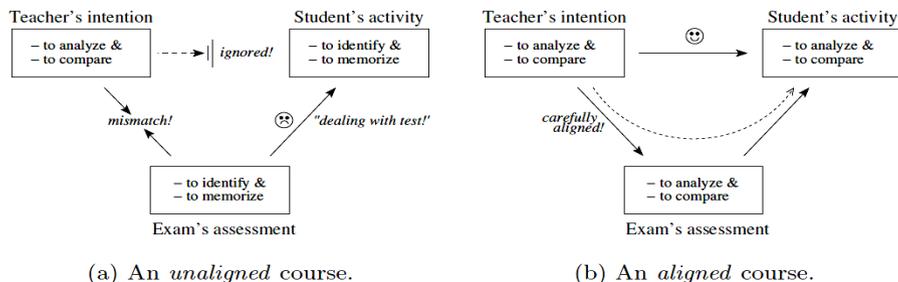


Abb. 2: Illustration der Notwendigkeit der Abstimmung von Kompetenz und Prüfung am Unterschied zwischen einem nicht abgestimmten und einem abgestimmten Kurs (aus [Br08])

4 Zur Formulierung von Kompetenzen

Sowohl aus didaktischer Sicht mit den Anforderungen des *Constructive Alignment* als auch aus hochschulpolitischer Sicht wird eine konkrete Formulierung des intendierten Outputs für jede Lehrveranstaltung gefordert. Ein Blick in die Lehrveranstaltungsbeschreibungen verschiedener Universitäten zeigt, dass dies selten outputorientiert ge-

² Für eine anschauliche Darstellung sei der Film *Teaching Teaching & Understanding Understanding* [BA06] empfohlen sowie [Bi03].

³ Begrifflichkeit angepasst. Biggs formuliert hier *intended learning outcomes*.

schieht.⁴ Die Begrifflichkeiten zur Formulierung von „Output“ sind noch nicht geklärt und werden unterschiedlich gehandhabt. Nach dem Entwurf des DQR bezeichnen Lernergebnisse (learning outcomes) das, „was Lernende wissen, verstehen und in der Lage sowie bereit sind zu tun, nachdem sie einen Lernprozess abgeschlossen haben.“ Lernergebnisse werden gebündelt und dann als Kompetenz beschrieben. Diese Verwendung des Begriffs Lernergebnis wird aber an anderer Stelle kritisiert (vgl. [BIB06]):

Die Definition des Begriffs Lernergebnis ist wenig fundiert und kaum operationalisierbar. Es wird vorgeschlagen, vom Lernergebnisbegriff abzurücken und auf Kompetenzen, ihre Entwicklung und Aneignung zu fokussieren. Was zählen soll, sind die erworbenen und nachweisbaren Kompetenzen, gerade unabhängig davon, ob sie nun in Bildungsinstitutionen, im Prozess der Arbeit, im sozialen Umfeld, in allgemeiner oder beruflicher Bildung, auf der Basis von in Lehrplänen und Lehrbüchern niedergelegten Wissensbeständen oder durch Erfahrung erworben wurden.

Da auch in der deutschen Informatikdidaktik bisher vor allem von Kompetenzen (im Sinne von intendierten Lernergebnissen) gesprochen wird, soll im Folgenden ausschließlich der Begriff Kompetenz im o.g. Verständnis verwendet werden. In der Planung und Durchführung von Lehrveranstaltungen handelt es sich damit um intendierte Kompetenzen, also Kompetenzen, welche die Studierenden nach der Lehrveranstaltung erworben haben sollten und in einer geeigneten Prüfung nachweisen müssen.

In verschiedenen nationalen und internationalen Publikationen finden sich Leitfäden und Richtlinien zur Formulierung intendierter Kompetenzen (z. B. [Ka03; GTW09]). Eine zentrale Stellung nimmt bei der Formulierung von Kompetenzen das Verb ein, welches angibt, was ein Studierender in der Lage ist zu *tun* bzw. *tat*, um die Kompetenz nachzuweisen. Ein Adverb gibt in dieser Verbindung an, wie gut oder in welcher Ausprägung dieser Nachweis erbracht wird. Entsprechend müssen Kompetenzen operationalisierbar sein, um eine Leistungserhebung zu ermöglichen.

In der Formulierung solcher Kompetenzen liegt die größte Schwierigkeit, wie [DB07] in einer Analyse von an die Bologna-Empfehlungen angepassten Studienprogrammen in den Niederlanden nachwies. Unzureichende Formulierungen von Kompetenzen resultieren in der Regel aus ungenauen, generalisierten und abstrakten Angaben und fehlenden Hinweisen dazu, wie solche Leistungen überprüft würden. Folgende Hauptprobleme wurden identifiziert (jeweils mit Beispiel, zitiert nach [Ad09]):

- Aussagen sind keine echten Kompetenzen:
“[A graduate is able to] Discuss in an informed manner the implications of professional regulation for nursing practice“.
Was als Aktivität beschrieben wird, sagt nichts darüber aus, was *“an informed manner“* bedeutet.
- Aussagen sind oft so ungenau, dass sie bedeutungslos werden:
“Graduates are able to apply the knowledge to solve qualitative and quantitative problems of a chemical nature“.
- Aussagen, die zwar weniger ungenau sind, aber nicht genau ausdrücken, was Absolventen eines Kurses können sollen:
“Graduates are able to conduct a whole range of laboratory procedures and use of instrumentation in synthetic and analytical work“.

⁴ Z.B. in der Beschreibung der Vorlesung „Einführung in die Programmierung“ an der Universität Paderborn: (<http://www.cs.uni-paderborn.de/fachgebiete/ag-wehrheim/lehre/ws0910/gp1.html>)

- Offensichtliche Aussagen:
 “*On successful completion students should be able to undertake appropriate further training or study of a professional or equivalent nature*”.

Ziel des Verbs in einer Kompetenzaussage ist also nicht nur die Angabe, wie eine Kompetenz überprüft wird, sondern auch ein Anhaltspunkt für die Studierenden, welche kognitiven Aktivitäten sie in einer Lehrveranstaltung bewältigen müssen, um erfolgreich abzuschließen. Je konkreter die durch das Verb transportierte Handlung, umso deutlicher wird es für die Studenten, was sie *tun* müssen, und desto genauer wissen die Lehrenden, wozu sie die Studenten befähigen müssen. Um die in einer Kompetenz innewohnende Aktivität zu verdeutlichen, empfiehlt Brabrand [Br08], Lehrveranstaltungsbeschreibungen zu formulieren nach dem Muster

„*After the course, students are expected to be able to ...*“

Die Verwendung typischer Verben für die Angabe von Kompetenzen für den Informatikunterricht und die Hochschulinformatik wird im folgenden Abschnitt analysiert.

Kompetenzen als Spezifikation

Es erstaunt angesichts gängiger Formalismen in der Informatik, dass für Informatiklehrveranstaltungen selten präzise *Spezifikationen* angegeben werden. Jede informatische Problemlösung beginnt mit der Definition des Problems selbst: Man erstellt eine sogenannte Spezifikation des Problems. Die outputorientierte Beschreibung einer Lehrveranstaltung stellt eine solche Spezifikation dar: Nach der Begrifflichkeit der *funktionalen Spezifikation* wird zum einen die Funktionalität festgelegt, welches in diesem Zusammenhang der Kompetenzentwicklung der Studenten entspricht. Zum anderen werden festgelegt:

- Die Vorbedingungen *pre (preconditions)*, welche die Einschränkungen festlegen; in diesem Kontext sind es die Kompetenzen, die Studenten zu Beginn einer Lehrveranstaltung besitzen müssen.
- Die Nachbedingungen *post (postconditions)*, welche die Kompetenzen der Studierenden beschreiben, die die Lehrveranstaltung erfolgreich abschließen.

Das Erreichen der *Funktionalität* wird durch vom Dozenten festzulegende Lernaktivitäten realisiert, die dann bei den Studenten zum Entwickeln der unter *post* festgelegten Kompetenzen führen, die schließlich durch eine adäquate Kompetenzfeststellung getestet werden.

Entsprechend dem Prinzip des *Constructive Alignment* müssen die Lernaktivitäten auf die Eingangskompetenzen K_{pre} , Eigenschaften der Studierenden (Motivation, Hintergrund, Interessen, ...) und die angestrebten Kompetenzen zugeschnitten sein. Die Prüfung muss die Lernaktivitäten widerspiegeln. Eine Lehrveranstaltung ist umso erfolgreicher, je mehr Studierende in der vorgesehenen Zeit durch die Lernaktivitäten die angestrebten Kompetenzen erreichen. K_{pre} und K_{post} müssen also klar formuliert und dargestellt werden.

5 Kompetenzformulierungen für Informatikunterricht und -lehre

Fachspezifische Kompetenzen sollten sich zwischen den verschiedenen Bildungsstufen zumindest partiell überdecken, so dass zunächst Formulierungen basierend auf den Vorarbeiten für den Informatikunterricht betrachtet werden.

Von der Arbeitsgruppe „Operatoren“ des 16. Fachdidaktischen Gesprächs zur Informatik in Königstein wurde ein erster Entwurf erstellt, der bei der Formulierung erwarteter Prüfungsleistungen im Informatikabitur behilflich sein soll und auf Operatorenlisten der Prüfungsanforderungen verschiedener Bundesländer basiert [KS09].

Folgende Operatoren werden für die Informatik aufgezählt:

Angaben, Begründen, Berechnen, Beschreiben, Bestimmen, Beurteilen, Bewerten, Darstellen, Entwerfen, Erläutern, Implementieren, Stellung nehmen, Überführen, Vergleichen.

An gleicher Stelle wird auch eine Liste von Verben angegeben, die nicht als Operator verwendet werden sollen, z.B.

Analysieren, Anwenden, Auswerten, Kommentieren, Modellieren.

Hierbei handelt es sich zum einen um Synonyme⁵ der Operatorenliste (angeben – nennen), zum anderen um Tätigkeiten, die sich nicht als Operator eignen, bspw. aufgrund der hohen Komplexität oder Uneindeutigkeit (z.B. *Modellieren*).

Zu jedem Operator sind Umschreibungen, eine Zuordnung zu einem Anforderungsbereich der EPA, Beispiele und Bemerkungen angegeben.

In den GI-Empfehlungen für Bildungsstandards Informatik finden sich Kompetenzformulierungen für die Sekundarstufe I, die als früher Versuch angesehen werden können, Kompetenzen für die Informatik zu formulieren. So finden sich hier auch verschiedene Formulierungen, die sich nach o.g. Argumentation nicht als Operator eignen, bspw.: „*Die SuS kennen und verwenden Baumstrukturen am Beispiel von Verzeichnisbäumen.*“ Aus diesen Formulierungen wird nicht deutlich, was konkret von den Schülern erwartet wird bzw. wie dieses zu prüfen ist. An solchen Formulierungen wird auch deutlich, dass die Formulierung von Kompetenzen nicht einfach aus Unterrichtsinhalten bzw. Lernzielen extrahierbar ist.

Kohl und Fothe [KF07] greifen in ihrem Kompetenzmodell für den Inhaltsbereich Algorithmen die in den GI-Empfehlungen formulierten Kompetenzen auf und formulieren diese für drei Kompetenzstufen detailliert aus. Diese können beispielhaft für die Formulierung weiterer Kompetenzen verwendet werden, z. B.: Die SuS

- fertigen einen schriftlichen Entwurf für Programme an,
- implementieren Programme mit einem Programmiersystem benutzungsfreundlich,
- testen Programme anhand gegebener Eingaben auf ihre Funktionalität.

[Po07] gibt einen Entwurf für ein Kompetenzstufenmodell basierend auf den Formulierungen der EPA und den Empfehlungen der Gesellschaft für Informatik e.V. an. Auch hier finden sich noch verschiedene Ausprägungen von Verben, die keine konkrete und prüfbare Handlung bezeichnen und damit sicherlich auch Auslöser dafür waren, dass das Erstellen einer Operatorenliste in Königstein aufgegriffen wurde. Gleichzeitig warnt Poloczek vor der Gefahr, Operatoren überzustrapazieren: Auch die vormals gewünschte kleinschrittige Überprüfbarkeit der Zielorientierung sei in der Praxis aufgegeben worden.

⁵ „Synonyme sollen nicht zugelassen werden, weil sie der Grundidee des Operatorkonzepts widersprechen“.

„Eine Verlagerung zu kompetenzorientiertem Unterricht hin verlangt vor der Einführung grundsätzliche Überlegungen zum Inhalt der Lehrpläne. Ansonsten besteht die Möglichkeit, dass die bisherigen Inhalte weiterhin den Mittelpunkt der Unterrichtsplanung bilden und damit den Kompetenzen nicht das notwendige Augenmerk gewidmet wird.“ Gleiches kann für die Hochschullehre angenommen werden.

Welche Kompetenzen sind für Informatikstudenten relevant? Aufgrund der höheren Komplexität der zu erwerbenden Kompetenzen im Informatikstudium ist hier eine größere Bandbreite an Kompetenzen zu erwarten. Bisher sind nur wenige Aufsätze zu Kompetenzen publiziert worden – vermutlich auch aufgrund der Komplexität eines solchen Unterfangens. Die Fachkommission Informatik der PHen in Baden Württemberg [BKM09] schlägt „Kompetenzen für das Lehramt Informatik“ vor. In Anlehnung an Fachbücher der Informatik geben sie einen Katalog von Kompetenzbereichen an, die für einen erfolgreichen Studienabschluss als notwendig erachtet werden; eine konkrete Formulierung der Kompetenzen wird nicht vorgenommen. In der vorliegenden Version stellt die Darstellung nur eine (weitere) Angabe von Inhalten dar.

Brabrand [Br08] gibt verschiedene Beispiele für outputorientierte Kursbeschreibungen aus dem Bereich der Informatik an (vgl. Abb. 3). Explizit wird darauf hingewiesen, dass das Auswendiglernen von Inhalten nicht zu den intendierten Lernergebnissen zählt.

- After the course, students are expected to be able to
- ~~memorize content~~
 - construct models from specifications;
 - apply standard solutions to common concurrency problems;
 - relate models and specifications;
 - test models wrt. behavior (using tool support);
 - define relevant safety/liveness properties for models;
 - verify models wrt. safety/liveness properties (using tools);
 - analyze models (and programs) wrt. behavior;
 - compare models (and programs) wrt. behavior;
 - implement models in familiar programming languages; and
 - relate models and implementations.

Abb. 3: Kompetenzformulierungen im Kurs „Model-Based Design for Concurrency“

6 Einordnung und Beurteilung der Kompetenzen

Ziel universitärer Lehrveranstaltungen ist es, Studierende zur Bewältigung möglichst komplexer kognitiver Aufgaben zu befähigen. Die Zuordnung der Kursbeschreibungen zu verschiedenen Kompetenzstufen einer adäquaten Taxonomie ermöglicht eine Überprüfung dieses Ziels. Die in der beschriebenen Weise explizit formulierten Kompetenzen lassen eine Einordnung in Lernzieltaxonomien zu. Die Mehrzahl existierender Lernzieltaxonomien wurde allerdings nicht vor dem Hintergrund der Hochschullehre entwickelt. Biggs [Bi03] gibt mit der SOLO-Taxonomie (SOLO – Structure of the Observed Learning Outcome) ein Strukturierungsinstrument an, welches explizit für die Hochschullehre entwickelt wurde und von Studenten erworbene Kompetenzen verschiedenen Kom-

plexitätsstufen zuordnet. Kompetenzsteigerung wird verstanden als das Erreichen von Kompetenzen einer höheren Stufe. Unterschieden werden *surface learning* (Stufen 2 und 3) und *deep learning* (Stufen 4 und 5), vgl. Abb. 4.

Durch den Fokus auf die Lernergebnisse kann die SOLO-Taxonomie herangezogen werden, die Komplexität intendierter Kompetenzen zuzuordnen. Auch [Ko09] ordnet die Kompetenzen seines Kompetenzmodells den verschiedenen SOLO-Stufen zu, um die Komplexität der verschiedenen Stufen analysieren zu können.

SOLO 2 <i>uni-structural</i>	SOLO 3 <i>multi-structural</i>	SOLO 4 <i>relational</i>	SOLO 5 <i>extended abstract</i>
<ul style="list-style-type: none"> - paraphrase - define - identify - count - name recite - follow (simple) instructions - ... 	<ul style="list-style-type: none"> - combine - classify - structure - describe - enumerate - list - do algorithm - apply method - ... 	<ul style="list-style-type: none"> - analyze - compare - contrast - integrate - relate - explain causes - apply theory (to its domain) - ... 	<ul style="list-style-type: none"> - theorize - generalize - hypothesize - predict - judge - transfer theory (to new domain) - ...

Abb. 4: Stufen 2–5 der SOLO-Taxonomie und korrespondierende Aktivitäten [Bi03]

An der Fachhochschule Aachen [SPK04] wurde ein Vokabular entwickelt, welches Kompetenzen in sechs verschiedenen Bereichen konkretisiert:

1. Aktivitäten, die Wissen nachweisen: (z.B. definieren, beschreiben, identifizieren)
2. Aktivitäten, die Verstehen nachweisen: (z.B. interpretieren, übersetzen, schätzen)
3. Aktivitäten, die Wissen/Verstehen nachweisen: (z.B. anwenden, lösen, erstellen)
4. Aktivitäten, die Analyse nachweisen: (z.B. erkennen, unterscheiden zwischen, bewerten)
5. Aktivitäten, die Synthese nachweisen: (z.B. vorschlagen, darstellen, strukturieren)
6. Aktivitäten, die Evaluation nachweisen: (z.B. beurteilen, bewerten, schließen, vergleichen)

Dieser Katalog kann zur Kompetenzformulierung und Kompetenzbewertung herangezogen werden.

7 Diskussion und Fazit

„Man mag zum Bachelor für die Informatik stehen, wie man will, aber eine positive Aussage ist die Beteuerung der Politik gewesen, in diesem Studienabschnitt die didaktischen Anstrengungen und die Betreuung und Beratung deutlich zu verstärken“ [C106]. Die Ausrichtung der Studienmodule im Grundstudium nach der Theorie des *Constructive Alignment* bietet die Chance, wie aus anderen Ländern berichtet, die Studienmotivation und Leistungen der Studierenden zu steigern.

Zusätzlich hilft die Angabe der zu erwartenden und geprüften Kompetenzen den Studenten, die Relevanz und Anwendbarkeit der Inhalte besser einzuschätzen. So werden im Grundstudium Informatik die Veranstaltungen zu den verschiedenen Teilgebieten der Informatik separat angeboten. Studienanfängern fällt es in diesem Stadium schwer, das

„große Bild“ bzw. die Zusammenhänge der Teildisziplinen in der Informatik zu verstehen. Innerhalb der Teilgebiete wird oftmals nicht deutlich, welche Aspekte der in der Vorlesung dargebotenen Inhalte nun tatsächlich relevant sind: Sowohl für das Verständnis, als auch für die Prüfung. Soll für das Fach Theoretische Informatik bspw. das „Pumping-Lemma“ bewiesen werden können, der Beweis nachvollzogen und erklärt werden können? Reicht es ggf. aus, Sinn und Ziele des Pumping-Lemmas zu erklären oder dieses anzuwenden? Oder diene das PL nur als Anschauungsmittel zur Verdeutlichung der Abgrenzung verschiedener Sprachklassen? Einem Studenten ist es unmöglich, allein einzuschätzen, was hier von ihm gefordert wird. Eine Konkretisierung der Kompetenzen anstelle der Aussage „Alles ist wichtig“ hilft den Studenten, die Relevanz der Inhalte einzuschätzen und die Lernorganisation darauf abzustimmen.

In welchem Umfang die Abstimmung umsetzbar ist und wie direkt sich diese auswirken wird, bleibt abzuwarten und zu untersuchen. An der Pädagogischen Hochschule Schwäbisch Gmünd werden wir mit dem WS 2010/11 schrittweise alle Lehrveranstaltungen des Instituts für Mathematik und Informatik auf eine outputorientierte transparente Darstellung der zu erwerbenden Kompetenzen umstellen und die Auswirkungen erheben. Hierzu wird von jedem Lehrenden eine Liste von Kompetenzen erarbeitet, die die Ziele der jeweiligen Lehrveranstaltung verdeutlichen. Mit den ersten Entwürfen beginnt die Phase der Diskussion und Ordnung, in welcher im Gesamtkanon der Studiengänge fehlende oder redundante Kompetenzen ermittelt und untereinander abgeglichen werden.

Literatur

- [Ad09] Adelman, C.: The Bologna Process for U.S. Eyes: Re-learning Higher Education in the Age of Convergence. Institute for Higher Education Policy, Washington, DC, 2009.
- [ADQ09] ADQ, Arbeitskreis Deutscher Qualifikationsrahmen: Diskussionsvorschlag eines Deutschen Qualifikationsrahmens für lebenslanges Lernen. 2009.
- [BIB06] BIBB, Bundesinstitut für Berufsbildung: Fachlicher Prüfbericht zu den Grundbegriffen und Deskriptoren des Entwurfs für einen Europäischen Qualifikationsrahmen. 2006. <http://www.bibb.de/de/25717.htm>
- [BKM09] Bieser, H.; Kortenkamp, U.; Morgen, M.; Müller, K.; Spannagel, C.; Zandler, A.: Kompetenzen für das Lehramt Informatik, unveröffentlichtes Dokument, 2009.
- [Bi03] Biggs, J.: Teaching for quality learning at university Society for Research into Higher Education and Open University Press, Berkshire, UK, 2003.
- [Bl06] Blum, W.: Die Bildungsstandards Mathematik – Einführung. In (Drücke-Noe, C.; Hartung, R.; Köller, O., Hrsg.): Bildungsstandards Mathematik: konkret, Berlin, Cornelsen-Scriptor-Verlag, 2006; 14-29.
- [Br08] Brabrand, C.: Constructive Alignment for Teaching Model-Based Design for Concurrency. In Transactions on Petri Nets and Other Models of Concurrency 5100/2008, 2008; S. 1-18.
- [BA06] Brabrand, C.; Andersen, J.: Teaching Teaching & Understanding Understanding. 2006. www.daimi.au.dk/~brabrand/short-film/
- [BD07] Brabrand, C.; Dahl, B.: Constructive alignment and the SOLO taxonomy: a comparative study of university competences in computer science vs. mathematics. Proc., 2007.
- [BD09] Brabrand, C.; Dahl, B.: Using the SOLO-Taxonomy to Analyze Competence Progression of University Science Curricula. In Higher Education 58(4), 2009; S. 531-549.
- [CI06] Claus, V.: 2 b v - 2 b - Maßnahmen zur Förderung der Hochschuldidaktik Informatik. Proc. HDI 2006, 2006.
- [DB07] de Bruin, et al: Competences in Education and Cross-Border Recognition: Evaluation of the Usefulness of Learning Outcomes and Competences for International Recognition. NUFFIC, The Hague, NL, 2007.

- [EK08] EU-Kommission: Der Europäische Qualifikationsrahmen für lebenslanges Lernen. 2008. http://ec.europa.eu/education/policies/educ/eqf/eqf08_de.pdf
- [GTW09] Grün, G.; Tritscher-Archan, S.; Weiß, S.: Leitfaden zur Beschreibung von Lernergebnissen. 2009. http://ibw4.m-services.at/zoom/pdf/wp2/Leitfaden_DE_final_2.pdf
- [Ka03] Kahn, P.: Guidance on writing aims and intended learning outcomes. 2003. <http://www.eps.manchester.ac.uk/.../guidance-on-aims-inteded-learning-outcomes.pdf>
- [Ko09] Kohl, L.: Kompetenzorientierter Informatikunterricht in der Sekundarstufe I unter Verwendung der visuellen Programmiersprache Puck. Jena, 2009.
- [KF07] Kohl, L.; Fothe, M.: Algorithmen aus einer anderen Perspektive. In LOG IN 146/147, 2007; S. 20-22.
- [KS09] KS, Arbeitsgruppe Operatoren Königstein: Ergebnisse der Arbeitsgruppe "Operatoren des 16. Fachdidaktischen Gesprächs zur Informatik in Königstein. 2009. www.imoodle.de/wikis/operators/index.php
- [Ni10] Nilson, L. B.: Teaching at its best: a research-based resource for college instructors Jossey-Bass, 2010.
- [Po07] Poloczek, J.: Kompetenzorientierter Informatikunterricht, Präsentation zur MNU-Tagung 2007. 2007. http://www.informatik.uni-frankfurt.de/~poloczek/MNU-Tagung-2007-09-27/kompetenzorientierter_Informatikunterricht.pdf
- [RS06] Romeike, R.; Schwill, A.: "Das Studium könnte zu schwierig für mich sein" - Zwischenergebnisse einer Langzeitbefragung zur Studienwahl Informatik. Proc. HDI 2006: Hochschuldidaktik der Informatik, München, Lecture Notes in Informatics, 2006.
- [Ve05] Veiga, A.: Europeanization of Higher Education Area: Towards a Framework of Analysis. Centre for Research in Higher Education Policies, Porto, PT, 2005.
- [WDW06] Weicker, N.; Draskoczy, B.; Weicker, K.: Fachintegrierte Vermittlung von Schlüsselkompetenzen der Informatik. Proc. HDI 2006: Hochschuldidaktik der Informatik, München, LNI, 2006.
- [WW08] Weicker, N.; Weicker, K.: Analyse des Kompetenzerwerbs im Softwarepraktikum. Proc. HDI 2008, Potsdam, Universitätsverlag Potsdam, 2008.
- [We02] Weinert, F. E.: Vergleichende Leistungsmessung in Schulen - eine umstrittene Selbstverständlichkeit. In (Weinert, F. E., Hrsg.): Leistungsmessungen in Schulen, Weinheim, 17-31, 2002.

Pseudo: eine Programmiersprache auf der Basis von Pseudocode zur Unterstützung der akademischen Lehre

Marcus Frenkel

Karsten Weicker

HTWK Leipzig, Fakultät IMN

04277 Leipzig

Web: portal.imn.htwk-leipzig.de/fakultaet/weicker

E-Mail: {mfrenkel,weicker}@imn.htwk-leipzig.de

Zusammenfassung: Pseudo ist eine auf Pseudocode basierende Programmiersprache, welche in der akademischen Lehre zum Einsatz kommen und hier die Vermittlung und Untersuchung von Algorithmen und Datenstrukturen unterstützen soll. Dieser Beitrag geht auf die Besonderheiten der Sprache sowie mögliche didaktische Szenarien ein.

Einleitung

Ein wichtiger Bestandteil in der Ausbildung von Informatikern ist die umfassende und verständliche Vermittlung von Algorithmen und Datenstrukturen in unterschiedlichen Veranstaltungen auf Bachelor- wie auch Masterniveau. Neben den klassischen asymptotischen Betrachtungen gewinnen auch empirische Untersuchungen immer stärker an Bedeutung – als Grundlage für effektive Designentscheidungen im Informatikeralltag. Das Ziel dieser Arbeit ist die Verknüpfung einer möglichst einfach lesbaren Notation mit einer direkten Ausführbarkeit der Algorithmen. Hiervon versprechen wir uns eine intensivere studentische Auseinandersetzung mit den Inhalten.

In der Lehre reicht die Notation von Algorithmen von rein textueller Beschreibung über oft individuell ausgeprägten Pseudocode hin zu echtem Programmcode – jede mit spezifischen Vor- und Nachteilen.

Textuelle Beschreibungen bieten den höchsten Grad an deskriptiver Freiheit, da sie keinem Formalismus unterworfen sind. Dies birgt den Nachteil einer oft abstrakten und mehrdeutigen Beschreibung. Weitere Nachteile sind hoher Platzbedarf und fehlende Möglichkeiten, die Korrektheit zu prüfen.

Programmcode hingegen bietet neben Eindeutigkeit und Präzision auch Möglichkeiten zur Überprüfung und weitergehenden Untersuchung, da der Code ausgeführt werden kann. In den meisten Programmiersprachen bedingen die strenge Syntax und die umfassenden programmiersprachlichen Konzepte (z.B. Objektorientierung, Ausnahmebehandlung etc.) allerdings eine stark aufgeblähte Beschreibung und verstellen den Blick auf den eigentlichen Algorithmus.

Pseudocode steht zwischen diesen beiden Extremen und geht einen Mittelweg zwischen Freiheit und Exaktheit. Durch mathematische Ausdrücke ergibt sich oft eine hohe Ausdrucksstärke und Kompaktheit, die in den beiden Alternativen nicht möglich ist. Aller-

dings übernimmt Pseudocode die fehlende Ausführbarkeit von textuellen Beschreibungen und auch Fehlinterpretationen sind dank der unterschiedlichen Handhabung von Referenzen bzw. formalen Parametern oder durch fehlende Typinformation, etwa bei der Anweisung $mitte \leftarrow (a+b)/2$, keine Seltenheit.

In der aktuellen Informatiklehre fehlt dementsprechend eine Möglichkeit zur geeigneten Notation von Algorithmen und Datenstrukturen, die sowohl einen genügend großen Detaillierungsgrad bei hoher Abstraktion bietet als auch eine automatische Verarbeitung durch einen Parser und Compiler erlaubt, um die syntaktische und semantische Korrektheit sowie Eindeutigkeit zu gewährleisten. Neben der reinen Ausführung des Codes sollen auch empirische Untersuchungen hinsichtlich Zeit- und Speicherverbrauch ermöglicht werden.

Pseudo

Um die in Kapitel 1 genannten Ziele einer geeigneten Notationsform für Algorithmen und Datenstrukturen zu erreichen, wurde eine neue Programmiersprache mit der Bezeichnung „Pseudo“ – in Anlehnung an den Pseudocode – entwickelt.

Pseudo orientiert sich an einer mathematischen Notation, da diese sowohl knapp, dabei aber auch präzise, ausdrucksstark und eindeutig ist. Da nicht davon auszugehen ist, dass häufig benutzte mathematische Symbole wie das Element-Zeichen \in , der Linkspfeil \leftarrow oder das Zeichen für Ungleichheit \neq über gängige Tastaturen eingegeben werden können, werden derartige Symbole über entsprechende Ersatzzeichen und Zeichenketten umschrieben. Das gesamte Erscheinungsbild der Sprache ist derart ausgelegt, dass durch einfaches Parsen und Konvertieren des Codes eine mathematische Notation zur Darstellung auf Folien und in Büchern und Artikeln erstellbar ist.

Als der Sprache zugrundeliegendes Programmierparadigma wurde eine Mischung aus imperativem und objektorientiertem Paradigma gewählt, da auf diese Art die Vorteile beider Paradigmen kombiniert und gleichzeitig ihre Nachteile vermieden werden können. Der objektorientierte Anteil ermöglicht die Umsetzung komplexer, benutzerdefinierter Typen und direkt damit verknüpfter Funktionen zur Erklärung von Datenstrukturen, wohingegen der imperative Anteil die Definition globaler Funktionen und damit die einfache Erklärung von Algorithmen erlaubt. Zusätzlich erlaubt Pseudo ähnlich wie C/C++ die Kontrolle über die Verwendung von Wert- beziehungsweise Referenzzuweisungen, wobei zwei verschiedene Zuweisungsarten – der Linkspfeil \leftarrow als Wertzuweisung und der Rechtspfeil \rightarrow als Referenzzuweisung, eingeführt wurden.

Ein weiterer wichtiger Aspekt von Pseudo ist die vorhandene strenge Typisierung. Dies erfordert zwar zusätzliche Syntaxelemente und erhöht den Platzaufwand geschriebenen Codes, trägt aber erheblich zum leichteren Verständnis dargestellter Algorithmen und Datenstrukturen bei, da die Typen verwendeter Variablen nicht mehr erraten oder abgeleitet werden müssen.

Im Rahmen der strengen Typisierung wurden für Pseudo verschiedene Standarddatentypen festgelegt, die zur Repräsentation von primitiven und zusammengesetzten Daten eingesetzt werden können. Neben üblichen skalaren Typen für natürliche, ganze und reelle Zahlen sowie Zeichenketten und Wahrheitswerte existieren als zusammengesetzte Typen Listen, Felder, Mengen und Tupel mit jeweils eigenen, an der Mathematik orientierten Konstruktoren.

Ein Beispiel für einen in die mathematische Darstellung konvertierten, in Pseudo geschriebenen Algorithmus bietet die folgende Abbildung an Hand von Quicksort:

```

1:   Quicksort( $\mathbb{T}$ )  $\in (a \in \mathbb{T}^2 \times l, r \in \mathbb{N}) \rightarrow ()$  :
2:       if  $l < r$  :
3:            $index \leftarrow Partition(a, l, r) \in \mathbb{N}$ ;
4:
5:           Quicksort( $a, l, index - 1$ );
6:           Quicksort( $a, index + 1, r$ );
7:
8:   Partition( $\mathbb{T}$ )  $\in (a \in \mathbb{T}^2 \times l, r \in \mathbb{N}) \rightarrow \mathbb{N}$  :
9:        $i \leftarrow l, j \leftarrow r \in \mathbb{N}$ ;
10:       $p \rightarrow a_i \in \mathbb{T}$ ;
11:
12:      while  $i < j$  :
13:          if  $a_j > p$  :            $j--$ ;
14:          else if  $a_i < p$  :  $i++$ ;
15:          else :                  $(a_i, a_j) \rightarrow (a_j, a_i)$ ;
16:
17:           $(a_i, a_j) \rightarrow (a_j, a_i)$ ;
18:      return  $j$ ;

```

Als weitere Besonderheit von Pseudo wird durch spracheigene Mittel die einfache Parallelisierung sowie die Untersuchung von Algorithmen durch Profiling ermöglicht. Die Realisierung dieser beiden Konzepte erfolgt direkt im Programmcode durch Annotationen.

Die Annotationen erlauben dem Benutzer, den eigentlichen Programmcode um zusätzliche Informationen anzureichern; sie werden durch eine eigene Syntaxstruktur beschrieben, die durch den Austausch einzelner Wörter innerhalb der Struktur den gewünschten Beschreibungseffekt hervorruft. Die Annotationen in Pseudo orientieren sich in ihrer Notation an der Sprache C#, werden also durch in eckige Klammern eingeschlossene Schlüsselwörter definiert, wobei zusätzliche Argumente in runden Klammern angegeben werden. Im Unterschied zu C# erlaubt Pseudo allerdings auch Annotationen direkt im Code einer Funktion, wodurch sich erweiterte Kontrollmöglichkeiten über die Art der Codeausführung ergeben. Neben Möglichkeiten zur Zugriffskontrolle auf Typbestandteile, Implementierungsinformationen für bestimmte Datenstrukturen wie etwa Mengen, Angaben zum Speichermanagement oder der Blockbildung sind die beiden für die Lehre wichtigsten Funktionen von Annotationen das Profiling und die Beschreibung von Parallelität im Programmcode.

Das Profiling erfasst unterschiedlichste Ablaufparameter, wie etwa die Anzahl von Funktionsaufrufen, die Anzahl ausgeführter Instruktionen, die für einen Funktionsaufruf aufgewendete Zeit oder den Speicherverbrauch einer Funktion, aber auch weitere Metriken wie die durchschnittliche Anzahl von Durchläufen einer Schleife oder einer Verzweigung, die Anzahl von Zugriffen auf eine Variable oder die Anzahl von erstellten Instanzen eines Typs. All diese Informationen können durch Pseudo während der Codeausführung ermittelt werden, wobei entweder der gesamte Programmcode oder speziell mar-

kierte Codebestandteile analysiert werden. Die Markierung der Codebestandteile erfolgt über die Annotation `profile`, welche im Zusammenhang mit dem Profiling an Funktions-, Typ- und Variablendeklarationen sowie an Kontrollstrukturen mit untergeordneten Blöcken (wie etwa Schleifen oder Verzweigungen) benutzbar ist. Argumente in der Annotation bestimmen die jeweils benutzten Metriken. Im folgenden Beispiel werden einige Anwendungsbereiche für die Profile-Annotation dargestellt, wobei für den Typ `T` die Anzahl der Instanzen sowie die Anzahl der Zugriffe auf die einzelnen Bestandteile, für die Funktion `g` die Anzahl der beim Aufruf ausgeführten Instruktionen und für die Verzweigung sämtliche verfügbare Metriken aufgezeichnet werden:

```

profile(instances, calls)
type T = ( n element N ) :
  f: () -> N :
    return n;

[profile(instructions)]
g: ( b element B ) -> T :
  t element T;
  [profile]
  if b = true :
    t.n <- 1;
  else :
    t.f();
  return t;

```

Neben der Untersuchung von Algorithmen ist die Beschreibung von Parallelität ein wichtiger Faktor in der Lehre. Gerade bei Algorithmen und Datenstrukturen kommt der Aspekt der Parallelisierung jedoch häufig zu kurz – trotz der steigenden Bedeutung durch die Mehrkernprozessoren. Die Lehre konzentriert sich dabei häufig auf abstrakte Konzepte oder komplexe Programmbibliotheken wie PVM oder MPI. Eine einfache, intuitiv verständliche und dabei dennoch umfangreiche Möglichkeit zur Beschreibung und vor allem auch Ausführung von Parallelität existiert bisher nicht. Daher erlaubt Pseudo die Deklaration von umfangreicher Parallelität mit Hilfe der Annotationen, ohne dabei auf die intuitive Verständlichkeit zu verzichten.

Mit Blick auf die gängigsten Formen von Parallelität werden in Pseudo synchrone und asynchrone Prozesse sowie Prozesse mit getrenntem und solche mit gemeinsamem Adressraum unterschieden. Die Angabe des gewünschten Modus der Parallelität wird über die Argumente der Annotation `parallel` erreicht; weitere Angaben zur grundlegenden Parallelität sind nicht nötig, da ein Mapping der benötigten Prozesse auf die verfügbaren Recheneinheiten automatisch ausgeführt wird. Um im Programmiermodell mit gemeinsamem Adressraum den konfliktfreien Informationsaustausch zu gewährleisten, stehen die Schlüsselwörter `lock` und `unlock` zum Sperren von Variablen zur Verfügung. Zusätzlich werden Prozesse blockiert, die eine bereits gesperrte Variable mit einer weiteren Sperre belegen wollen; die Blockierung wird erst aufgehoben, wenn der erste sperrende Prozess seine Sperre aufhebt (Deadlock-Situationen müssen also immer noch durch den Programmierer vermieden werden). Die Prozesskommunikation im verteilten Adressraum erfolgt dagegen über die Schlüsselwörter `send`, `receive` und `scatter`; die Identifizierung bestimmter Prozesse erfolgt hierbei über die automatisch durch den Compiler global definierten Variablen `processID`, `processIDs` und `activeProcesses`.

Die `parallel`-Annotation kann im Zusammenhang mit bedingten Verzweigungen und der Blockbildung benutzt werden, wobei hierbei jede Anweisung im der Annotation untergeordneten Block in einem separaten Prozess ausgeführt wird. Der folgende Code zeigt zwei kurze Beispiele:

```

parallelFunction: () -> () :
  a = ( i, j ) element T^2;
  mainProcess <- processID element ProcessID;

  // 1: synchron, gemeinsamer Adressraum
  [parallel]
    i <- f();
    j <- g();

  // 2: synchron, getrennter Adressraum
  [parallel(distinct)]
    if processID = processIDs.last()[0] :
      i <- f();
      receive j from processIDs.last()[1];

    if processID = processIDs.last()[1] :
      send g() to processIDs.last()[0];

```

Neben Blöcken und bedingten Anweisungen lassen sich auch Iteratorschleifen parallelisieren, wobei für jedes Element der Quelldatenmenge der Schleifencode in einem separaten Prozess ausgeführt wird. Durch die Angabe eines scatter-Argumentes können zusätzlich angegebene weitere Datenmengen auf die verschiedenen Prozesse aufgeteilt werden, um so eine gleichmäßige Bearbeitung größerer Datenmengen durch mehrere Prozesse zu erlauben.

Didaktischer Wert und Fazit

Welcher didaktische Mehrwert entsteht nun daraus, dass in ansprechender, knapper und präziser Form beschriebene Algorithmen direkt mit Beispielen ausführbar sind? Anhand weniger Lehr-/Lernszenarien soll dies kurz illustriert werden.

In der reinen Präsenzlehre bietet der ausführbare Pseudo-Code eine höhere Anschaulichkeit, da mit kleinen Beispielen Details wie das Umhängen von Zeigern bei einer Baumrotation oder die Berechnungsreihenfolge bei den optimalen Suchbäumen als direkter Effekt des langsam ablaufenden Algorithmus erlebbar werden. Selbst mit reinen Textausgaben von Zwischenständen wird der Ablauf des abstrakt formulierten Algorithmus „erlebbar“ – wünschenswert ist zukünftig jedoch eine zusätzliche Visualisierungskomponente.

Für Übungsaufgaben bietet Pseudo alternative Aufgabenstellungen: Durch das Profiling von Algorithmen (z.B. Quicksort und Bottom-Up-Heapsort) oder Datenstrukturen (z.B. AVL- und Rot/Schwarz-Bäume) bzgl. der Laufzeit (oder der Anzahl der Schlüsselvergleiche oder Lese-/Schreibzugriffe) können Studenten direkt die Eigenschaften eines ablaufenden Algorithmus mit den Ergebnissen der asymptotischen Analyse vergleichen. Bisher war eine derartige Aufgabenstellung nur durch erheblichen, nicht mit einem Mehrwert für die Lehre einhergehenden zusätzlichen Programmieraufwand möglich. Das Experimentieren mit Algorithmen zielt auf die Kompetenz ab, die Techniken später in „echten“ Anwendungen einsetzen und verschiedene Ansätze im Praxiseinsatz gegeneinander abwägen zu können. Auch die Erstellung von Varianten einer Datenstruktur (z.B. bei B-Bäumen) kann leicht und spielerisch von Studenten bewältigt werden.

Darüber hinaus ermöglicht der Ansatz von Pseudo als vollständige Programmiersprache den leichten Einsatz und die Kombination von Datenstrukturen sowie Algorithmen; so können schnell kleine Anwendungsszenarien von den Studierenden durchgespielt wer-

den. Diese Synthese-Kompetenz in realistischen Problemstellungen ist bisher bei den Studierenden eher schwach ausgeprägt.

Ein weiterer wichtiger didaktischer Aspekt wird vor dem Hintergrund der zunehmenden Anzahl der Prozessorkerne deutlich – eine Herausforderung, der die heutigen Programmierer kaum gewachsen sind [PT08]. In Anbetracht dieses grundlegenden Wandels in der alltäglichen Softwareentwicklung bietet Pseudo mit seiner annotationsgesteuerten Parallelisierung einige einfache, aber grundlegende Konzepte für die Vermittlung von Parallelisierung im selben Namensraum – erneut mit der Möglichkeit, die Basistechniken durch Experimentieren selbst zu erfahren.

Den Autoren ist kein anderer Ansatz bekannt, der leichte Lesbarkeit (z.B. eines Pseudocodes) mit der direkten Ausführbarkeit (und einem Profiling) verbindet. Zur Beschreibung von Algorithmen geht am ehesten Knuths MIXAL auf einer maschinennahen Ebene einen ähnlichen Weg [K97]; auch eine Beschreibung von Algorithmen durch eine funktionale Sprache kann natürlich einige Aspekte ähnlich gut erfüllen.

Wo steht das Pseudo derzeit? Die Konzeption der Sprache ist abgeschlossen und es liegt eine prototypische Implementation vor. Für eine vollständige Entwicklungs- und Simulatorumgebung liegt derzeit das Hauptaugenmerk auf der Umsetzung des Profiling und der Parallelität sowie dem Export in eine mathematische Notation, da hier die großen Stärken von Pseudo liegen.

Hinsichtlich Nutzen und Akzeptanz liegen noch keine empirischen Daten vor, es ist jedoch geplant, die Sprache Pseudo im Sommersemester 2011 in größerem Umfang in einer Lehrveranstaltung des zweiten Fachsemesters einzusetzen. Wir werden dies durch eine entsprechende Befragung der Studierenden beziehungsweise gegebenenfalls auch durch eine Vergleichsstudie begleiten. Im Sommersemester 2010 wurden bereits einzelne Details der Pseudo-Notation ohne eine Werkzeugunterstützung an den Studierenden „erprobt“. Naturgemäß wird Pseudo durch Rückmeldungen beim Lehreinsatz Anpassungen im Sprachdesign unterliegen. In diesem Zuge soll auch geprüft werden, ob andere Lehrinhalte, wie etwa Entwurfsmuster, einfach in Pseudo realisierbar sind. Für eine größere Verbreitung bei Lehrenden kann auch die Definition eigener Operatorsymbole und Syntaxstrukturen geprüft werden.

Literatur

- [PT08] V. Pankratius, W. F. Tichy: Die Multicore-Revolution und ihre Bedeutung für die Softwareentwicklung, Objektspektrum 2008
- [K97] D. E. Knuth: Art of Computer Programming, Volume 1: Fundamental Algorithms (3rd Edition), 3rd ed. Addison-Wesley Professional, 1997

Integration eines Rapid-Feedback-Moduls in eine koaktive Lern- und Arbeitsumgebung

**Wolfgang Reinhardt, Michael Mascher,
Senol Gül und Johannes Magenheim**

Institut für Informatik – Universität Paderborn
33102 Paderborn – Germany
Web: ddi.upb.de
E-Mail: {wolle,mima,senolg,jsm}@upb.de

Zusammenfassung: Die Evaluierung von Lehrveranstaltungen hat in vielen Lehreinrichtungen eine lange Tradition. In diesen klassischen Evaluierungsszenarien werden einmalig pro Semester Umfragebögen an die Studierenden verteilt und anschließend manuell ausgewertet. Die Ergebnisse sind dann zumeist am Ende der Vorlesungszeit vorhanden und geben einen punktuellen Einblick in die Qualität der Lehrveranstaltung bis zum Zeitpunkt der durchgeführten Evaluation. In diesem Artikel stellen wir das Konzept des Rapid Feedback, seine Einsatzmöglichkeiten in universitären Lehrveranstaltungen und eine prototypische Integration in eine koaktive Lern- und Arbeitsumgebung vor.

1 Einführung

Verfolgt man die öffentliche Diskussion, die in den letzten Jahren über Lehreinrichtungen geführt wurden, so stößt man immer wieder auf den Aspekt der Qualitätssicherung von Lehrveranstaltungen [Kro03,RM08]. Sowohl die Institution Schule als auch Fachhochschulen und Universitäten beschäftigen sich sehr intensiv mit etwaigen Methoden, um die Qualität der Lehre zu sichern und nachhaltig zu verbessern. So ist es beispielsweise an Universitäten üblich, dass Lehrveranstaltungen anhand von Fragebögen evaluiert werden. Die Ergebnisse liefern Rückschlüsse darauf, inwiefern die Rahmenbedingungen (Größe des Raumes, technische Ausstattung etc.) und die inhaltlichen Merkmale (logischer Aufbau der Veranstaltung, Schwierigkeitsgrad etc.) den Vorstellungen und Wünschen der Studierenden entsprechen. Zudem erhalten die an der Veranstaltung beteiligten Lehrpersonen eine Rückmeldung über ihre geleistete Arbeit und können dies nutzen, um ihre Fertigkeiten im Bereich der Lehre zu hinterfragen und ggf. zu verbessern.

Neben der Möglichkeit, Evaluationen mit Hilfe von Fragebögen durchzuführen, gibt es eine ganze Reihe alternativer Befragungstechniken. Vor allem Internet und WWW stellen hierbei eine überaus reizvolle Plattform für die Durchführung von Umfragen und Evaluationen dar, da sie mit der Verbreitung des Mediums an Bedeutung und Beliebtheit gewinnen. Diese Beliebtheit erklärt sich zusätzlich dadurch, dass sie eine zeitlich effiziente und kostengünstige Form der Datengewinnung und -verarbeitung darstellen.

Der vorliegende Artikel ist wie folgt gegliedert: Abschnitt 2 stellt Technologie-gestützte Methoden der Evaluierung und Qualitätssicherung in Lehrveranstaltungen vor und führt in das Konzept des Rapid Feedbacks ein. Die technischen Grundlagen und Erweiterungsmöglichkeiten des koaktiven Lernmanagementsystems koaLA sowie die Konzeption und Implementierung des Rapid-Feedback-Moduls werden in Abschnitt 3 vorgestellt.

Abschnitt 4 fasst den Artikel zusammen und zeigt Möglichkeiten für eine zukünftige Weiterentwicklung und praktische Erprobung im größeren Kontext auf.

2 Evaluation und Qualitätssicherung im universitären Kontext

Die wohlbekannten klassischen Umfragemethoden zur Evaluation wie die persönliche, telefonische oder postalische Befragung [BWGB99] sollen in diesem Artikel nicht berücksichtigt werden, da die erstgenannten in universitären (Massen-)Veranstaltungen nur sehr selten zum Einsatz kommen, der postalische Befragungsweg mit hohen Kosten verbunden ist und darüber hinaus im Rahmen dieses Artikels eine technische Integration von Befragungsmethoden in ein Lernmanagementsystem angestrebt wird.

2.1 Technologie-gestützte Evaluationsmöglichkeiten

Innerhalb der empirischen Sozialforschung stellt die Methode der Technologie-gestützten Befragung (oder auch Onlinebefragung) eine relativ neue Alternative zu den traditionellen Umfrageformen dar. Aufgrund der stark wachsenden Bedeutung des Mediums Internet für die Gesellschaft versuchen sowohl die Sozial- als auch die Marktforschung diese Plattform ebenfalls für ihre Zwecke zu nutzen. *E-Mail-Befragungen* im Internet durchzuführen ist eine naheliegende Variante von Internet-basierten Umfragen. Hierfür wird an eine ganze Reihe von E-Mail-Adressen ein Fragebogen gesandt, der von dem Probanden ausgefüllt und wieder zurückgeschickt wird. Aufgrund verschiedener organisatorischer Restriktionen besitzt diese Form der Befragung allerdings keine bedeutende Relevanz bei Befragungen mit Hilfe des Internets. Auch *Newsgroup-Befragungen* gehören zur Kategorie der Technologie-gestützten Evaluationsmöglichkeiten. Bei dieser Methode werden die Fragen in einer bestimmten Gruppe veröffentlicht und von den dort anwesenden Benutzern beantwortet. Newsgroups gehören heutzutage zu den kaum noch verwendeten Internet-Diensten, weshalb eine Umfrage, die mit Hilfe von Newsgroups verteilt wird, nur eine sehr geringe Rücklaufquote findet. Berücksichtigt man die angesprochenen Probleme der anderen beiden Varianten einer Onlineumfrage, so ist es nicht verwunderlich, dass *WWW-Befragungen* die am häufigsten verwendete Form einer Internet-Befragung sind. Bei dieser Befragungsmethode werden, technisch unterstützt durch spezialisierte Umfragewerkzeuge oder einfache HTML-Formulare, Fragebögen von den Probanden innerhalb eines Webbrowsers ausgefüllt. [BWGB99] nennen jedoch das Auffinden eines Fragebogens im WWW als einen Nachteil von WWW-Befragungen, da wohl kein Proband auf die Suche nach einer Befragung geht, sondern der Link zu einer solchen Befragung wiederum über einen anderen Kommunikationskanal zum Probanden kommen muss.

2.2 Rapid Feedback

Der hier vorgestellte Rapid-Feedback-Ansatz zur Evaluation von Lehrveranstaltungen unterstützt klassische Evaluationsansätze und kombiniert diese mit Technologie-gestützten Evaluationsmöglichkeiten. Dabei werden die Vorteile so genannter Web-2.0-Techniken und -Technologien gewinnbringend eingesetzt, um sowohl Studierende wie auch Lehrende während der Evaluation optimal zu unterstützen.

„Rapid Feedback bezeichnet ein Konzept aus der empirischen Onlineforschung mit dem Ziel klassische Umfrageformen zur Qualitätssicherung in Lehrveranstaltungen durch Technologie-gestützte Ansätze zu unterstützen. Dazu werden beim Rapid Feedback Onlineumfragen durchgeführt, die es dem Lehrenden sehr schnell und einfach ermöglichen inhaltliche und organisatorische Sachverhalte bei den Lernern zu erfragen. Rapid bezieht sich dabei sowohl auf die Erstellung und die Durchführung, als auch auf die Auswertung von Umfragen. Die durchgeführten Umfragen sollten dabei möglichst einfach und klar strukturiert sein, um eine automatisierte Auswertung zu ermöglichen.“

Als Rapid Feedback verstehen wir also Onlinebefragungen – genauer gesagt WWW-Befragungen –, die den Studierenden schnell und ohne Nutzungsbarrieren zugänglich gemacht und von den Lehrenden einfach und ohne großen Aufwand ausgewertet werden können. Um die Nutzungsbarrieren so gering wie möglich zu halten und Medienbrüche zu minimieren, sind Rapid Feedbacks idealerweise in das eingesetzte LMS integriert, um so die vorhandenen Zugangs- und Rechteschemata zu nutzen und so die Befragung auf die Grundgesamtheit der Teilnehmer der Lehrveranstaltung zu begrenzen. Die Bekanntmachung einer Befragung kann über die integrierten Kommunikationskanäle des LMS erfolgen und den Studierenden stets am Ort der Lehrmaterialien und sonstigen Lehrszenarien präsentiert werden. Fragebögen, die zum Rapid Feedback verwendet werden, sollten sich als Vorlagen speichern lassen und zeitgesteuert aktiviert und deaktiviert werden können, um so den organisatorischen Aufwand für die Lehrenden zu optimieren. Um den Lehrenden die Möglichkeit zur zeitnahen Reaktion auf die Ergebnisse einer Befragung unter den Studierenden ihrer Lehrveranstaltung zu ermöglichen, sollte eine Rapid-Feedback-Anwendung eine (deskriptive) Auswertung in Echtzeit ermöglichen. Für statistische Auswertungen in spezialisierten Softwareanwendungen sollten entsprechende Schnittstellen existieren. Wie in allen anderen Evaluationsmethoden auch, ist es beim Rapid Feedback sehr wichtig, dass der Durchführende einer Befragung diese mit dem Ziel der Verbesserung der eigenen Lehrqualität durchführt. Die Verbesserungspotentiale betreffen dabei sowohl den Vermittlungsprozess als auch die eingesetzten Materialien und Hilfsmittel und nicht zuletzt auch den Lehrenden selbst.

2.3 Evaluation in universitären Lehrveranstaltungen

Eine Analyse der Bedingungen innerhalb von Lehrveranstaltungen fördert sehr schnell zu Tage, dass Onlineumfragen sehr gut für den Einsatz innerhalb von Lehrveranstaltungen geeignet sind. Dabei ist vernachlässigbar, ob es sich bei der Bildungseinrichtung um eine Universität oder eine Schule und es sich dementsprechend um Seminar-, Vorlesungs- oder Projektgruppen oder um Schulklassen handelt. Die Menge der Grundgesamtheit, die für die Umfrage interessant ist, ist in jedem Fall überschaubar und vor allem eindeutig abgrenzbar. Einschränkend sei hierzu bemerkt, dass diese Voraussetzung hauptsächlich in Schulklassen sowie Seminaren und Projektgruppen an Universitäten eindeutig gegeben ist, da hier die Zahlen der Personen in den jeweiligen Gruppen bereits im Voraus bekannt sind. Handelt es sich bei der Lehrveranstaltung um eine Vorlesung, ist die Bestimmung aller Teilnehmer deutlich komplizierter. Insbesondere ist problematisch zu gewährleisten, dass nur die Personen an der Evaluation teilnehmen, die die Lehrveranstaltung wirklich besucht haben und somit in der Lage sind, diese auch wirklich zu beurteilen.

Funktionen von Lehr-Evaluationen

Die Untersuchung und Beurteilung eines bestimmten Untersuchungsgegenstands muss wissenschaftlich begründet und unter Anwendung wissenschaftlicher Methoden in Hinblick auf seinen Nutzen und seinen Wert erfolgen. Betrachtet man den hohen zeitlichen und ggf. finanziellen Aufwand einer Evaluation, stellt sich die Frage, welchen Zweck und welche Ziele die Evaluation einer Lehrveranstaltung verfolgt. Im Bildungskontext steht dabei insbesondere die Verbesserung der Qualität der Lehre im Vordergrund. Hierbei lassen sich vier Grundfunktionen von Lehr-Evaluationen differenzieren:

- *Entwicklung und Verbesserung*: Defizite in der Qualität der Lehre können durch die Erfassung des Ist-Zustands identifiziert und durch einen Vergleich mit erwarteten Ergebnissen eine gezielte Verbesserung ermöglichen.
- *Unterstützung bei Entscheidungssituationen*: Mittels Befragungen können Informationen erfasst werden, die in einer Entscheidungssituationen der Abwägung des Für und Wider dienlich sind. Häufig handelt sich es um Entscheidungen, die sich direkt auf den evaluierten Gegenstand beziehen, wie z.B. die Wiederholung eines Modellversuchs.
- *Rechenschaft und Legitimation*: Bereits durchgeführte Maßnahmen können mit Hilfe von Evaluationen gegenüber Dritten legitimiert werden. Außerdem kann das Verhalten einer weisungsgebundenen Person überprüft werden.
- *Lernen*: In dieser Funktion werden die Befragungsergebnisse indirekt verwendet, indem Informationen zur Effizienz und Wirkweise des untersuchten Gegenstands gewonnen werden.

Diese Grundfunktionen können im Rahmen einer Evaluation gleichzeitig sowie in unterschiedlichen Beziehungen zueinander verfolgt werden. Sie sind nicht als ausschließliche und isolierte Funktionen zu verstehen. Ausgehend von den hier beschriebenen allgemeinen Funktionen von Lehr-Evaluationen können konkrete Möglichkeiten für Rapid-Feedback-Befragungen in Seminaren, Vorlesungen und Projektgruppen im Hochschulkontext abgeleitet werden.

3 Konzeption und Entwicklung des Rapid Feedback Moduls

Im Rahmen des BMBF-geförderten Projekts Locomotion wurde an der Universität Paderborn die koaktive Lern- und Arbeitsplattform koaLA geschaffen, die klassische Funktionen des Lernmanagements mit Web-2.0-Funktionalitäten verbindet und selbst organisiere Lern- und Arbeitsräume bietet [RSB⁺07]. Das Projekt und die Entwicklung von koaLA waren von der Fragestellung geleitet, wie sich die inhaltlichen und technologischen Konzepte des so genannten Web 2.0 auf technische Infrastrukturen im universitären Kontext übertragen lassen können. koaLA ist seit dem Wintersemester 2007/08 im alltäglichen Praxiseinsatz an der Universität Paderborn und verbindet dabei traditionelle und informelle Lernkontexte für Studierende und Lehrende. Technisch basiert koaLA auf der Open-Source-Umgebung open-sTeam, die den Aufbau und die Pflege virtueller Wissensräume unterstützt und grundlegende Funktionen kooperativen Arbeitens und Lernens über Programmierschnittstellen bereitstellt. koaLA stellt darüber hinaus ein Erweiterungskonzept für Lernszenarien bereit, über welches die bereits implementierten kooperativen Funktionen um neue erweitert werden können [RSB⁺07].

3.1 Anforderungen an das Rapid Feedback Modul für koaLA

In Abschnitt 2.2 haben wir das Konzept des Rapid Feedbacks eingeführt und bereits funktionelle Anforderungen an eine Rapid-Feedback-Anwendung skizziert. Für die Entwicklung eines Rapid-Feedback-Moduls für koaLA können die Anforderungen in drei wesentliche Teilbereiche aufgeteilt werden: 1) Erstellung und Administration von Befragungen, 2) Durchführung von Befragungen und 3) Auswertung und Visualisierung der abgegebenen Antworten.

Das zu entwickelnde Modul muss sich an den vorhandenen technischen und visuellen Spezifikationen von koaLA orientieren und gängige Web-Standards implementieren. Hinsichtlich der Bedienbarkeit soll sich das Rapid-Feedback-Modul an der von Google Forms orientieren und somit den Benutzer mit einer schlanken Oberfläche auf die inhaltliche Ausgestaltung der Befragung fokussieren lassen. Die Untersuchung der an der Universität Paderborn derzeit eingesetzten Papier-basierten Evaluationsbögen ergab die Anforderung an 5 umzusetzende Fragetypen, um diese Bögen 1:1 nachbilden zu können: 1) einzeilige offene Frage, 2) mehrzeilige offene Frage, 3) geschlossene Frage mit einer Antwortmöglichkeit, 4) geschlossene Frage mit mehreren Antwortmöglichkeiten und 5) eine Matrix-Frage mit einer Antwortmöglichkeit pro Zeile.

Vor der Durchführung einer Befragung soll der Lehrende die Möglichkeit zur Vorschau der Befragung haben und eine Befragung als Vorlage für die spätere Wiederverwendung speichern können. Die Durchführung von Rapid Feedbacks soll manuell und zeitgesteuert aktivier- und deaktivierbar sein. Der Durchführende soll durch einen visuellen Hinweis auf den Status seiner Befragung hingewiesen werden. Für die Studierenden sollen die anstehenden Befragungen in einem separaten Wissensraum abgelegt werden und die mehrfache Teilnahme eines Studenten an ein und derselben Befragung muss systemseitig verhindert werden. Die erfolgreich gespeicherten Antworten sollen dem Durchführenden in einer automatisch erstellten deskriptiven Statistik inkl. entsprechender Visualisierungen präsentiert werden. Für weitergehende Auswertungen sollen entsprechende Exportformate bereitgestellt werden.

3.2 Umsetzung des Rapid Feedback Moduls für koaLA

Über den Extension-Manager von koaLA können Erweiterungen erfasst und registriert, aktiviert und deaktiviert werden. Er sorgt dafür, dass bei einem bestimmten Objekt-Typ die richtige Erweiterung angesprochen wird und koaLA die spezifischen Darstellungsoptionen der jeweiligen Erweiterung fehlerfrei darstellt und einbindet. Das Rapid-Feedback-Modul wurde als eine koaLA-Extension unter Verwendung der PHP-sTeam-Programmierschnittstelle entwickelt und fügt sich dank der Verwendung der koaLA-CSS-Gestaltungsdefinitionen nahtlos in das System ein. Sobald die Erweiterung serverseitig aktiviert wurde, können Rapid Feedbacks als neue Wissensräume (Lektionen) in den Veranstaltungsräumen angelegt werden.

Die Erstellung einer neuen Befragung orientiert sich am Erstellungsprozess im weit verbreiteten Umfragewerkzeug Google Forms. Für eine Befragung müssen ein Titel und Texte für die Willkommens- und Abschlusseite vergeben werden. Anschließend können in einer AJAX-basierten Oberfläche neue Fragen zur Befragung hinzugefügt werden. Die Reihenfolge der Fragen lässt sich durch Drag-and-Drop verändern und ermöglicht den Durchführenden einen sehr einfachen und intuitiven Erstellungsprozess für neue Befragungen.

4 Zusammenfassung und Ausblick

Qualitätsfeststellung, qualitätssichernde Maßnahmen und regelmäßige Evaluation von Lehrpraxis entwickeln sich zunehmend zu einem wichtigen Instrument des universitären Alltags. Dabei lassen sich vier verschiedene Grundfunktionen von Lehr-Evaluationen unterscheiden: 1) Entwicklung und Verbesserung von Lehrpraxis, 2) Unterstützung von Entscheidungssituationen, 3) Rechenschaft und Legitimation sowie 4) Lernen. In diesem Artikel haben wir klassische und Technologie-gestützte Evaluationsmethoden im universitären Kontext betrachtet und ihre jeweiligen Einsatzmöglichkeiten und Vorteile diskutiert. Mit dem Konzept des Rapid Feedbacks haben wir eine WWW-Befragung eingeführt, die in ein LMS eingebettet Möglichkeiten zur schnellen und einfach auswertbaren Befragung in Lehrveranstaltungen theoretisch ermöglicht. Weiterhin haben wir drei Lernszenarien vorgestellt, in denen sowohl Lehrende wie auch Studierende Vorteile durch den Einsatz einer Rapid-Feedback-Anwendung haben können.

Die konzipierte und implementierte Rapid-Feedback-Erweiterung für die koaktive Lern- und Arbeitsumgebung koaLA stellt für den Benutzer der Plattform, und ganz besonders den Lehrenden, eine komfortable und zugleich schnelle Möglichkeit dar, auf eigene Bedürfnisse angepasste Befragungen zu erstellen. Der Benutzer kann die Umfragen beliebig konfigurieren und durch Anpassen der Sichtbarkeit steuern. Die implementierten Fragetypen bieten eine Vielzahl von Umfrage-Variationen und die sofortige deskriptive und visuelle Auswertung ermöglicht das rasche Widerspiegeln von Ergebnissen an die Studierenden. Das entwickelte Rapid-Feedback-Modul für koaLA wird im Wintersemester 2010/11 erstmalig im Pilotbetrieb an der Universität Paderborn eingesetzt werden und somit wertvolles Feedback von Anwendern mit sich bringen. Bis zum planmäßigen Betrieb zum Sommersemester 2011 soll das Modul um weitere Fragetypen erweitert und mit einer fortgeschrittenen Auswertungsfunktionalität versehen werden.

Literatur

- [BWGB99] Bernard Batinic, Andreas Werner, Lorenz Gräf, Wolfgang Bandilla (Hrsg.): *Online Research, Methoden, Anwendungen und Ergebnisse*, Hogrefe, 1999.
- [Kro03] Helmut Kromrey: *Qualität und Evaluation im System Hochschule*. In: Reinhard Stockmann (Hrsg.): *Evaluationsforschung*, Leske+Budrich, 2003.
- [RM08] Wolfgang Reinhardt, Johannes Magenheimer: *Modulares Konzept für die Tutorenschulung in der universitären Informatikausbildung*. In: Andreas Schwill (Hrsg.): *Tagungsband zur Konferenz Hochschuldidaktik Informatik (HDI) 2008*, 2008.
- [RSB⁺07] Alexander Roth, René Sprotte, Daniel Büse, Thorsten Hampel: koaLA – Integrierte Lern- und Arbeitswelten für die Universität 2.0. In: 5. DeLFI, 2007.
- [Sch07] Wolfgang Schöning (Hrsg.): *Spuren der Schulevaluation – Zur Bedeutung und Wirksamkeit von Evaluationskonzepten im Schulalltag*, Klinkhardt, 2007.

Aquadrohne, Messdatenerfassung und Co. – Interdisziplinäres Projektmanagement als Teil des Wirtschaftsinformatikstudiums

Prof. Dr. Stephan Raimer

Fachbereich Wirtschaftsinformatik – Berufsakademie an der Wirtschaftsakademie
24106 Kiel - Germany

Web: www.wak-sh.de/berufsakademie.html

E-Mail: stephan.raimer@wak-sh.de

Zusammenfassung: Projektmanagement-Kompetenzen werden von Unternehmen unterschiedlichster Branchen mit wachsender Priorität betrachtet und eingefordert. Als Beitrag zu einer kompetenzorientierten Ausbildung werden in diesem Paper interdisziplinäre Studienmodule als Bestandteil des Wirtschaftsinformatik-Studiums vorgestellt. Zielsetzung der Studienmodule ist die Befähigung der Studierenden, konkrete Projekte unter Nutzung von standardisierten Werkzeugen und Methoden nach dem IPMA-Standard planen und durchführen zu können.

1 Bedeutung und Relevanz von Projektmanagement in der Informatik

Im Jahr 2007 präsentierte Deutsche Bank Research mit der Studie „Deutschland im Jahr 2020 – Neue Herausforderungen für ein Land auf Expedition“ [DBR07] Entwicklungspfade für die Wirtschaft und Gesellschaft in Deutschland. Kernaussage dieser Studie ist die Prognose, dass bis 2015 die *Projektwirtschaft* einen 15%-Anteil an der Wertschöpfung in Deutschland haben wird und hierbei besonders der deutsche Mittelstand einen Anteil haben soll. Unter der Projektwirtschaft werden „zumeist temporäre, außerordentlich kooperative und oft globale Wertschöpfungsprozesse“ verstanden, die „flexible Kooperation von Spezialisten“ [DBR07].

Neben dieser allgemeinen Perspektive wurden für den Bereich der IT-Projekte die von der „Standish Group“ (<http://www.standishgroup.com>) seit 1994 jährlich vorgelegten „Chaos Reports“ vielfach zitiert, kommentiert und zuletzt kritisiert. Wenngleich Zahlen wie ein Scheitern von 70% aller IT-Projekte, Projektabbruchraten um 30% und regelmäßige Budgetüberschreitungen mit Glass (2005) und Eman/Koru (2008) als deutlich überhöht angesehen werden müssen, bleibt der Bedarf an Projektmanagement-Kompetenzen in der IT-Wirtschaft eindrucksvoll belegt.

Zu den Erfolgsfaktoren für Projektmanagement in Unternehmen gehören hierbei klare Zielsetzung und gute Kommunikation sowie starke und in die Organisation integrierte Projektleiter (GPM/PACG 2008). Aktuelle Forschungsfragen im Kontext des Projektmanagements befassen sich etwa mit der Projektplanung, Werkzeugen, agilen Vorgehensmodellen [GPM09] oder der Messung der Komplexität in Projekten (s. etwa [P09]).

1.1 Projektmanagement im Wirtschaftsinformatik-Studium

Die eingangs geschilderte Situation in der Wirtschaft stellt umfangreiche Anforderungen an die Ausbildung in Hochschulen, insbesondere im Hinblick auf die Berücksichtigung von Projektmanagement-Kompetenzen.

Die Rahmenempfehlungen für das Studium in Wirtschaftsinformatik nach der Überführung des Ausbildungssystems von Diplom-Studiengängen zu Bachelor- und Masterprogrammen von 2007 [WI07] enthalten klare Aussagen zur Bedeutung von Projektmanagement-Kompetenzen. Zum einen wird unter den Gegenständen und Zielen der Ausbildung explizit im Kontext der Schlüsselqualifikationen auf die „Arbeit in interdisziplinären und ggf. verteilten Projektteams, auch länder-/kontinentübergreifend“ verwiesen, außerdem wird die konzeptionell-methodische Fundierung bei gleichzeitiger Berufs- und Arbeitsmarktorientierung des Studiums in den Vordergrund gestellt.

Weiterhin wird bei den in den Rahmenempfehlungen aufgeführten Hauptausbildungsbereichen der Wirtschaftsinformatik unter dem Punkt „Entwicklung und Management von Informationssystemen“ Projektmanagement als Inhaltsbereich aufgeführt.

2 Projektmanagement als Studienmodule

Die Berufsakademie an der Wirtschaftsakademie Schleswig-Holstein hat im Jahr 2006 die dualen Studiengänge der Bereiche Betriebswirtschaftslehre, Wirtschaftsingenieurwesen und Wirtschaftsinformatik von Diplom auf Bachelorabschlüsse umgestellt. Hierbei wurde für das Studium Wirtschaftsinformatik sowie Wirtschaftsingenieurwesen ein Pflichtmodul „Projektmanagement und Organisation“ (4. Semester) sowie ein Wahlpflichtmodul „Projektmanagement und Business Consulting“ (6. Semester) entwickelt. Diese Studienmodule wurden von der GPM (Gesellschaft für Projektmanagement e.V.) mit dem Zertifikat „Center of Excellence“ ausgezeichnet. Das in den Studienmodulen angestrebte Niveau ist dabei auf Projekte mit einer begrenzten Komplexität beschränkt.

Die neben den Studienleistungen optional angebotene Zertifizierung als Projektmanagement-Fachmann (IPMA-GPM Level D) wird dabei mittlerweile von deutlich mehr als der Hälfte der Studierenden eines Jahrgangs absolviert. Das Angebot wird von Studierenden wie auch von den Kooperationsunternehmen geschätzt und nachgefragt.

Referenz für die Studienmodule und Grundlage für die Zertifizierung bildet die *NCB National Competence Baseline 3.0* der *IPMA International Project Management Association* [NCB3]. Über die NCB3.0 werden Anforderungsprofile auf vier Niveaustufen für eine Zertifizierung nach EN ISO/IEC 17024 definiert. Professionelles Projektmanagement wird somit in drei Kompetenzbereiche gegliedert, die wiederum 46 Kompetenzelemente beinhalten.

Als Zielsetzung der Studienmodule wird entsprechend dem IPMA Level D eine Kernkompetenz in allen der nachfolgenden Kompetenzelemente gefordert, eine Anwendung steht im Vordergrund. (Ein zertifizierter Projektmanagement-Fachmann „kann die Kompetenzelemente anwenden“ und „arbeitet als Mitglied eines Projektteams oder gehört zum Projektpersonal.“ [NCB3, S.30])

PM-TECHNISCHE KOMPETENZELEMENTE	PM-VERHALTENS- KOMPETENZELEMENTE	PM-KONTEXT- KOMPETENZELEMENTE
1.01 Projektmanagementerfolg	2.01 Führung	3.01 Projektorientierung
1.02 Interessierte Parteien	2.02 Engagement und Motivation	3.02 Programmorientierung
1.03 Projektanforderungen und Projektziele	2.03 Selbststeuerung	3.03 Portfolioorientierung
1.04 Risiken und Chancen	2.04 Durchsetzungsvermögen	3.04 Einführung von Projekt-, Programm- und Portfoliomanagement
1.05 Qualität	2.05 Entspannung und Stressbewältigung	3.05 Stammorganisation
1.06 Projektorganisation	2.06 Offenheit	3.06 Geschäft
1.07 Teamarbeit	2.07 Kreativität	3.07 Systeme, Produkte und Technologie
1.08 Problemlösung	2.08 Ergebnisorientierung	3.08 Personalmanagement
1.09 Projektstrukturen	2.09 Effizienz	3.09 Gesundheit, Arbeits-, Betriebs- und Umweltschutz
1.10 Leistungsumfang und Lieferobjekte (Deliverables)	2.10 Beratung	3.10 Finanzierung
1.11 Projektphasen, Ablauf und Termine	2.11 Verhandlungen	3.11 Rechtliche Aspekte
1.12 Ressourcen	2.12 Konflikte und Krisen	
1.13 Kosten und Finanzmittel	2.13 Verlässlichkeit	
1.14 Beschaffung und Verträge	2.14 Wertschätzung	
1.15 Änderungen	2.15 Ethik	
1.16 Überwachung und Steuerung, Berichtswesen		
1.17 Information und Dokumentation		
1.18 Kommunikation		
1.19 Projektstart		
1.20 Projektabschluss		

Abb. 1: Kompetenzbereiche PM-Elemente NCB 3.0
 (Quelle: http://www.gpm-ipma.de/fileadmin/user_upload/Qualifizierung/Zertifizierung/Zertifikate_fuer_PM/ICB3_Elemente_gross.gif)

Zur Beurteilung der Kompetenz wird durch die NCB3.0 eine Taxonomie in einer 10-stufigen Skala definiert, diese ist so konzipiert „dass sie die berufliche Kompetenz der Personen in der praktischen Anwendung von Projektmanagement in ihrer Gesamtheit erfassen“ [NCB3, S.38]. Je nach angestrebtem Zertifizierungslevel werden unterschiedliche Niveaustufen gefordert (aufbauend vom IPMA-Level D).

2.1 Interdisziplinäre Projektarbeit

Bei der Realisierung der Studienmodule (mit 6 bzw. 8 Semesterwochenstunden) wird auf eine starken Praxisbezug Wert gelegt und die Umsetzung von Projekten in Teams von 2–3 Studierenden in den Vordergrund gestellt. Für diese Projekte wird pro Teammitglied jeweils ein Zeitaufwand von 20 Zeitstunden zugrunde gelegt und verschiedene Aufgabenstellungen (etwa zur Entwicklung von Konzepten, Machbarkeitsstudien und/oder Prototypen für Informationssysteme) bearbeitet.

Als Weiterentwicklung und Vertiefung für diese Projektarbeit wird für das 6. Semester ein interdisziplinäres Wahlpflichtmodul „Automatisierungstechnik/Technische Informatik“ (8 Semesterwochenstunden) angeboten, in dem Teams aus den Studienbereichen Wirtschaftsinformatik und Wirtschaftsingenieurwesen gemeinsam Projekte bearbeiten.

Im folgenden werden zwei Projektgegenstände exemplarisch vorgestellt, die Inhalte und Umfang der interdisziplinären studentischen Projekte aufzeigen sollen. Die Projekte wurden dabei teilweise arbeitsteilig durchgeführt und bauen iterativ aufeinander auf, außerdem zielen Sie auf Kernkompetenzen der beiden Studiengänge Wirtschaftsinformatik

und Wirtschaftsingenieurwesen. Im Sinne des Projektanspruches wurde auf Neuartigkeit und (begrenzte) Komplexität der Aufgabenstellung geachtet, bei zeitintensiven Anforderungen wurde eine arbeitsteilige Umsetzung (mehrere Projektteams) realisiert.

1. Projekt Aquadrohne

Zielsetzung war die arbeitsteilige Konstruktion und Fertigung einer Unterwasserdrohne in Kleinserie, die durch einen Mikrocontroller gesteuert autark Steuerprogramme abfährt – unter Einhaltung eines gesetzten Budgetrahmens.

Während für Studierende des Fachbereiches Wirtschaftsingenieurwesen die Gehäusekonstruktion, Dimensionierung von Akkukapazitäten und Antrieb sowie die elektrische Verschaltung von Sensoren und Aktoren im Vordergrund stand, beinhalteten die Zielsetzungen für Studierende des Fachbereiches Wirtschaftsinformatik die Auswahl des Mikrocontrollers inklusive Entwicklungsumgebung sowie der Entwurf und die Dokumentation des Steuerprogramms.



Abb. 2: Aquadrohne

Für die Realisierung der Steuerung wurde ein Arduino-Mega-Mikrocontroller (<http://arduino.cc/en/Main/ArduinoBoardMega>) ausgewählt, die Entwicklungsumgebung (<http://arduino.cc/en/Main/Software>) ist wie die Hardware als Open-Source-Lösung verfügbar. Das Steuerprogramm beinhaltet neben den Steuerungsaufgaben die Erkennung von Notfallsituationen, die zum Abbruch und Auftauchen führen (Eindringen von Wasser, Mangelnde Akkukapazität, Tauchtiefe zu groß).

2. Projekt Messdatenerfassung Wind-Solarkraft-Anlage

Zielsetzung war die Erfassung von Leistungsdaten einer kleinen Windkraftanlage sowie von Solarpanels, Auswahl eines geeigneten Übertragungsverfahrens und Visualisierung der Leistungsdaten als Webseite unter Beachtung eines Budgetrahmens.

Studierende aus dem Fachbereich Wirtschaftsingenieurwesen hatten hier den Schwerpunkt des Schaltungsaufbaus (Auswahl eines verlustoptimierten Verfahrens zur Messung der Leistungsparameter, Aufbau der Schaltung) während der IT-Anteil vor allem in der Auswahl einer Übertragungsverfahrens sowie der serverseitigen Visualisierung der Daten lag.



Abb. 3: Wind-Solarkraft-Anlage (vor Projektbeginn)

Für die Realisierung wurden zwei Übertragungsverfahren evaluiert, zunächst per WLAN-Funkstrecke, dann mit einem an einen Mikrocontroller gekoppelten GSM-Modul. Die letztlich gewählte Lösung setzt erneut auf einen Arduino-Mikrocontroller mit GSM-Shield, der mehrfach pro Stunde die Leistungsparameter per HTTP-Post an eine API der Plattform Pachube (<http://www.pachube.com/>) übergibt.

3 Zusammenfassung und Ausblick

Die vorgestellten Projektbeispiele wurden von den studentischen Teams erfolgreich fertiggestellt und gemäß den zuvor dargestellten Kompetenzelementen in einem Transfernachweis (mit Projektzielen, Umfeldanalyse, Risikoanalyse, Projektorganisation, Phasenplanung, Projektstrukturplan, Ablauf- und Terminplanung sowie Einsatzmittel- und Kostenplanung, Verhaltenskompetenz und einem Wahlelement in einem Umfang bis zu 60 Seiten) geplant, durchgeführt und dokumentiert.

Ergänzungen- und Weiterentwicklungen sind für nachfolgende Projektgruppen schon vorgesehen:

- Erstellung einer mobilen manuellen Fernsteuerung für die Aquadrohne per Zig-Bee, Integration zusätzlicher Sensoren sowie einer Kamera
- Erstellung von Alarmfunktionen (etwa SMS bei Sturm) für die Wind-Solarkraft-Anlage und zusätzliche Visualisierungen (etwa für mobile Endgeräte) sowie zusätzliche Auswertungsfunktionen

Die Besonderheit der Bearbeitung in der Bearbeitung der Projekte war die interdisziplinäre Ausrichtung, die Aufgabenstellungen konnten nur gemeinsam, unter Einbeziehung

und Anwendung spezifischer Fachkompetenzen (z.B. Softwareentwicklung vs. Elektrotechnik) gelöst werden. Gleichzeitig wurden in der Bearbeitung der Projekte Zusammenhänge zwischen Hard- und Software hergestellt und ein Einblick in spezifische Kenntnisse und Kompetenzen des jeweils anderen Studienganges gewonnen.

Aus der Perspektive der Lehrenden zeichneten sich die Studienmodule dadurch aus, dass die Projektteams ihre Aufgabenstellungen sehr motiviert und engagiert absolvierten.

Eine weitere Intensivierung der Zusammenarbeit in Form von zusätzlichen Wahlpflichtmodulen wird angestrebt, außerdem werden die vorhandenen Studienmodule durch weitere Projektthemen ausgebaut (etwa im Bereich der speicherprogrammierbaren Steuerungen).

Literatur

- [DBR07] Deutsche Bank Research: Deutschland im Jahr 2020 – Neue Herausforderungen für ein Land auf Expedition, 23. April 2007, http://www.dbresearch.de/PROD/DBR_INTERNET_DE-PROD/PROD000000000209595.pdf
- [EK08] Khaled El Emam, A. G. Koru: A Replicated Survey of IT Software Project Failures, IEEE Software, pp. 84–90, September/October, 2008
- [G05] Robert L. Glass: IT Failure Rates–70 Percent or 10–15 Percent?, IEEE Software, pp. 112, 110–111, May/June, 2005
- [GPM/PACG08] Studie der GPM und der PA Consulting Group: Erfolg und Scheitern im Projektmanagement, http://www.gpm-ipma.de/fileadmin/user_upload/Know-How/Ergebnisse_Erfolg_und_Scheitern-Studie_2008.pdf
- [GPM09] Konrad Spang, Sinan Özcan: GPM-Studie 2008/2009 zum Stand und Trend des Projektmanagements, http://www.gpm-ipma.de/fileadmin/user_upload/Know-How/00-Gesamt-Studie-GPM-Juli_2009.pdf
- [NCB3] GPM Deutsche Gesellschaft für Projektmanagement e.V. (Hrsg.): ICB – IPMA Competence Baseline – in der Fassung als Deutsche NCB – National Competence Baseline Version 3.0, März 2008, http://www.gpm-ipma.de/fileadmin/user_upload/Qualifizierung___Zertifizierung/Zertifikate_fuer_PM/NCB3_FINAL_20090912.pdf
- [P09] Gerold Patzak: Messung der Komplexität von Projekten, Projekt Management aktuell S. 42–45, 5/2009
- [WI07] Fachkommission im Auftrag der Wissenschaftlichen Kommission Wirtschaftsinformatik – Gesellschaft für Informatik: Rahmenempfehlungen für die Universitätsausbildung in Wirtschaftsinformatik (2007). In Kurbel u.a.: „Studienführer Wirtschaftsinformatik 2009 | 2010“, S. 19–37, Gabler, 2009

Ein virtueller Lernraum für die Informatiklehrerweiterbildung

Malte Dünnebier, Ira Diethelm

Universität Oldenburg
Department für Informatik, Abteilung Informatik in der Bildung
26111 Oldenburg - Germany
Web: www.ifib.uni-oldenburg.de
E-Mail: malte.duennebier@uni-oldenburg.de, ira.diethelm@uni-oldenburg.de

Zusammenfassung: Bei der Suche nach Möglichkeiten, die Weiterbildung für Informatiklehrkräfte auszubauen, bietet sich der Einsatz virtueller Lernräume an. Dieses Papier berichtet über ein Projekt, in dem ein exemplarischer virtueller Lernraum für kollaboratives Lernen in der Lehrerweiterbildung in Informatik theoriegeleitet erstellt, erprobt und bewertet wurde. Die erzielten Ergebnisse über das Nutzungsverhalten können für weitere E-Learningprojekte in der Lehrerbildung hilfreich sein. Der Schwerpunkt dieses Papiers liegt auf der Gestaltung des Lernraums unter Beachtung der speziellen Situation der Informatiklehrkräfte, nicht auf der didaktischen Aufbereitung der betreffenden Lerneinheit.

1 Motivation

Für eine flächendeckende Unterrichtsversorgung im Fach Informatik in der Sek. I bedarf es einer entsprechenden Anzahl von qualifizierten Lehrkräften. Leider stehen in den meisten Bundesländern derzeit gar nicht die notwendige Anzahl Lehrerinnen und Lehrer für die flächendeckende Einführung von Informatikunterricht zur Verfügung (vgl. [Mo07] S. 15). Zur Erhöhung der Zahl qualifizierter Informatiklehrkräfte ist neben der verstärkten grundständigen Ausbildung von Informatiklehrkräften an den Hochschulen die Weiterqualifizierung von bereits praktizierenden Lehrkräften denkbar. Hierbei ergibt sich in Flächenländern wie Niedersachsen oder Mecklenburg-Vorpommern, die nur über wenige diesbezügliche Weiterbildungsangebote verfügen, u.a. das Problem, dass es für die weiterbildungswilligen Lehrkräfte aufgrund der großen Entfernungen zu diesen Angeboten schwierig ist, an regelmäßigen Präsenzveranstaltungen teilzunehmen, bei welchen sie sich mit anderen Kollegen austauschen und gemeinsam lernen können. Außerdem sind die an ihren Schulen oft allein oder haben sehr wenige Informatikkollegen, von denen noch weniger grundständig ausgebildet sind. Doch Kommunikation und Kooperation ist für die eigene Weiterqualifizierung der Fachlehrkräfte und Weiterbildung fachfremder Lehrkräfte sehr wichtig. Dies betont u.a. Fussangel [Fu08] in ihrer Forschungsarbeit. Laut Fussangel sind insbesondere solche Gemeinschaften für eine nachhaltige Fort- und Weiterbildung von Lehrkräften geeignet, bei welchen die Lehrer „voneinander lernen, indem sie ihr individuelles Wissen aufeinander beziehen und sich gemeinsam neue Wissensbereiche erschließen“ ([Fu08], S. 264).

2 Aufbau des Lernraums für die Informatiklehrerweiterbildung

Vor dem Hintergrund der E-Learning-Debatte erscheint die Bereitstellung und Nutzung eines virtuellen Lernraumes als adäquates Mittel, der oben beschriebenen Problematik beizukommen (ein Beispiele hierfür [FLIEG]). Es stellt sich allerdings die Frage, wie ein solcher Lernraum für berufstätige Lehrkräfte der Sek. I zweckdienlich gestaltet sein sollte, wie diese spezielle Zielgruppe beim gemeinschaftlichen Arbeiten und Lernen in einem virtuellen Lernraum unterstützt werden kann. Bei der Suche nach entsprechenden grundlegenden Lehr-Lern-Konzepten im E-Learning-Kontext, gelangt man z. B. zur gemäßigt konstruktivistischen Lehre (vgl. u.a. [MGR95]). Nach dieser Lehr-Lerntheorie erfolgt Lernen in einem situativ eingebundenen, aktiven, konstruktiven, sozialen und selbst gesteuerten Prozess (vgl. u.a. [MGR95]). Ein Raum für „Computer Supported Collaborative Learning“ (CSCL) erscheint geeignet, dieses Lehr-Lern-Konzept in einem virtuellen Lernraum umzusetzen (siehe hierzu [Dü09], S.21 ff.). CSCL-Lernräume können mit verschiedenen Lernplattformen verwirklicht werden. Diese müssen hierfür insbesondere Werkzeuge anbieten, mit deren Hilfe die Lernenden in dem virtuellen Lernraum kommunizieren sowie möglichst asynchron und synchron an gemeinsamen Lernobjekten arbeiten können (vgl. [Sc03], S. 13).

Für die beispielhafte Erprobung eines solchen virtuellen Lernraumes für die Informatiklehrerweiterbildung in einem Flächenland bot sich die Lehrveranstaltung „Daten und Algorithmen“ des Lehrerweiterbildungs-Studienganges im Beifach Informatik der Universität Rostock an. Der Ablauf und die Inhalte des Lernmoduls waren weitestgehend von der Universität Rostock festgeschrieben. Der virtuelle Lernraum wurde unter Verwendung der Lernplattform Moodle exemplarisch für zwei Selbstlernphasen dieses Moduls konzipiert, die im Herbst 2009 stattgefunden haben. Eine initiale Adressatenanalyse ergab, dass die 32köpfige Lerngruppe (hiervon nahmen 30 an der Adressatenanalyse teil) sich im Gros als normale (60%) bis fortgeschrittene (34%) Computeranwender einschätzten, jedoch wenig Kontakt mit Online-Lernplattformen und virtuellen Lernräumen hatten: Nur 12% gaben an, bereits vor dem Projekt des öfteren eine Online-Lernplattform genutzt zu haben. Dementsprechend gering waren die Vorerfahrungen mit CSCL-Werkzeugen: Etwa 33% der Lerngruppenmitglieder hatte vor der Erprobung des virtuellen Lernraumes bereits entweder Chats (12 Teilnehmer) und/oder Messenger (10 Teilnehmer) und/oder Foren (10 Teilnehmer) aktiv genutzt. Im Umgang mit Gruppenkalendern hatten nur drei, mit Wikis nur zwei der Teilnehmer entsprechende Vorerfahrungen. Die Nutzung von E-Mails war hingegen fast allen Teilnehmern (29) vertraut.

Die Tatsache, dass viele Teilnehmer noch keine Vorerfahrungen im Umgang mit geläufigen CSCL-Werkzeugen hatten, erschwerte die Konzeption eines adressatenspezifischen CSCL-Lernraumes. Wäre bei der Ausgestaltung des Lernraumes nur auf CSCL-Werkzeuge zurückgegriffen worden, mit deren Nutzung die meisten Teilnehmer bereits im Vorfeld der Erprobung vertraut waren, wäre nur der Einsatz von E-Mails ohne Nutzung einer Online-Lernplattform möglich gewesen. Die Teilnehmer wurden deshalb in die Benutzung der Lernplattform, des virtuellen Lernraumes und der wichtigsten darin integrierten CSCL-Werkzeuge in der ersten Präsenzveranstaltung des Weiterbildungsstudienganges eingeführt. Darüber hinaus wurden entsprechende allgemeine Arbeits- und Informationsmaterialien in den Lernraum integriert.

Um den Kursteilnehmern in diesem virtuellen Lernraum kooperatives bzw. kollaboratives Lernen zu ermöglichen, wurden folgende CSCL-Werkzeuge eingesetzt:

Synchrone Kommunikationswerkzeuge: Der Moodle-Messenger, ein kursübergreifender Chat und Gruppenchats (Gruppenmodus: getrennte Gruppen).

Asynchrone Kommunikationswerkzeuge: Ein allgemeines Forum für alle Kursteilnehmer, Gruppenforen zur Kommunikation in den Lerngruppen.

Werkzeuge für Gruppenkoordination: Der Moodle-Kalender zur gemeinschaftlichen und persönlichen Terminkoordination.

Werkzeuge für kollaboratives Lernen und Arbeiten: Gruppenwikis, die für die Mitglieder der jeweiligen Teilgruppen sowie die Dozenten einseh- und editierbar waren. In diesen Wikis sollten die Kursteilnehmer in ihren Teilgruppen kooperativ bzw. möglichst kollaborativ die Arbeitsergebnisse zu den im Zuge der Selbstlernphasen gestellten Lernaufgaben erstellen.

3 Bewertung des Lernraums

Für die Erhebung der kommunikativen und kooperativen Aktivitäten der Kursteilnehmer in dem virtuellen Lernraum wurden u.a. die im Erhebungszeitraum gesammelten Moodle-Logdaten herangezogen. Diese Logdaten lassen allerdings keine Aussagen über die Qualität von Kommunikations- und Kooperationsaktivitäten der Kursteilnehmer zu, da z. B. die Länge und der Inhalt von Foren- oder Chatbeiträgen vom System dort nicht hinterlegt werden. Dennoch konnte mit Hilfe dieser Daten festgestellt werden, dass das CSCL-Werkzeug Chat von den Kursteilnehmern nur wenig und von vielen Teilnehmern gar nicht genutzt wurde: Nach Abzug einer einzigen intensiveren Chatsitzung zwischen zwei Teilnehmern, war keine nennenswerte Nutzung dieses CSCL-Werkzeuges mehr erkennbar. Dies lässt den Schluss zu, dass Chats für diese exemplarische Zielgruppe und diesen speziellen Einsatzkontext nicht geeignet waren, kooperatives bzw. kollaboratives Lernen in dem erprobten virtuellen Lernraum als Kommunikationswerkzeug zu unterstützen, denn auch Teilnehmer, die sich selbst als erfahrene Chatter beurteilt hatten, haben dieses Instrument nicht genutzt.

Die in den Lernraum integrierten Foren wurden laut Logdaten grundsätzlich besser angenommen: Insgesamt sechs der 32 Teilnehmer hatten das allgemeine Forum genutzt. Hier von hatten drei laut eigener Aussage bereits zuvor Erfahrungen mit Foren gesammelt. Die Gruppenforen der ersten Distance-Learning-Phase wurden von allen Gruppen zum Informationsaustausch genutzt. Bei den Gruppenforen der zweiten Selbstlernphase war die Gesamtbeteiligung verhältnismäßig gering (15 Forenbeiträge) und es war häufig nur eine unidirektionale Kommunikation in den Logdaten zu erkennen. Diese Entwicklung kann ggf. auf einen Neuigkeitseffekt oder andere gruppendynamische Prozesse zurückgeführt werden.

Obwohl fast alle Kursteilnehmer keine Vorerfahrung mit der aktiven Nutzung von Wikis hatten, fiel es ihnen anscheinend nicht schwer, sich in den Umgang mit den bereitgestellten Wikis einzuarbeiten: Die Anzahl der Wiki-Bearbeitungen (Logdateneintrag „wiki-edit“) der zugehörigen Gruppenwikis (in Summe 61) und deren Verteilung auf die Teilgruppenmitglieder ließen darauf schließen, dass die meisten Arbeitsgruppen ihr Wiki in der ersten Distance-Learning-Phase dazu genutzt haben, die Lösung zu der entsprechenden Aufgabe kooperativ zu erstellen. In den meisten Gruppen (7 von 11) haben mehrere Gruppenmitglieder dieses Wiki ein oder mehrmals bearbeitet. In der zweiten Fernlernphase wurde das zugehörige Gruppenwiki jedoch nur sehr zögerlich, in manchen Gruppen gar nicht oder nicht kooperativ genutzt. Die Datenlage ließ nur bei drei von elf Gruppen eine kooperative Erstellung des Wikis erkennen.

Dieses CSCL-Werkzeug scheint sich laut Logdaten-Interpretation also bei dieser speziellen Zielgruppe und dem gewählten Erprobungskontext zwar grundsätzlich für das ko-

operative bzw. kollaborative webgestützte Bearbeiten von Lernaufgaben geeignet zu haben, bot sich in der zweiten Distance-Learning-Phase aber offensichtlich für die Kursteilnehmer größtenteils nicht mehr zum kooperativen Bearbeiten der Aufgabenstellungen an.

Über diese Auswertung der Logdaten hinaus wurden die subjektiven Eindrücke der Teilnehmer bzgl. des virtuellen Lernraumes erhoben. Hierfür wurde ein Online-Fragebogen entworfen. Nachfolgend sind die Fragen aufgeführt, auf deren Antwortdaten im Wesentlichen in diesem Beitrag Bezug genommen wird:

Frage 1: Haben Sie sich in dem Moodle-Kursraum zurecht gefunden?

Antworttyp: geschlossen, ordinale Skala mit vier Merkmalen: nein; nach längerer Einarbeitungsphase (mehr als 3 Besuche notwendig); nach kurzer Einarbeitungsphase (weniger als 3 Besuche notwendig); ja, auf Anhieb.

Frage 3: Fanden Sie die Gruppenwikis bei der gemeinsamen Bearbeitung der Lernaufgaben in Ihrer Gruppe hilfreich?

Antworttyp: geschlossen. Ordinale Skala mit vier Merkmalen: nein; eher weniger; einigermaßen; ja.

Frage 6: Welche der folgenden Werkzeuge fanden Sie praktisch, um mit ihren Gruppenmitgliedern zu kommunizieren?

Antworttyp: geschlossen. Mehrfachantworten möglich. Nominale Skala mit sechs Merkmalen: keines der angeführten; E-Mail; Gruppenchats; Sofortmitteilungen; Gruppenforen; Gruppenwikis.

Frage 7: Welche der folgenden Werkzeuge fanden Sie praktisch, um mit allen Kursmitgliedern (auch keine Gruppenmitglieder) zu kommunizieren?

Antworttyp: geschlossen. Mehrfachantworten möglich. Nominale Skala mit fünf Merkmalen; keines der angeführten; E-Mail; allgemeiner Chat; Sofortmitteilungen; allgemeines Forum.

Frage 9: Was würden Sie an dem Kurs-Lernraum verbessern?

Antworttyp: offen.

Ein Großteil der Befragten bewertete den Aufbau des virtuellen Lernraumes grundsätzlich positiv: 29% gaben an, dass sie sich schnell und gut in dem Lernraum orientieren konnten, 42% benötigten lediglich eine kurze Einarbeitungsphase. Die Antworten derjenigen Lernenden auf die erste Frage der Abschlussbefragung, die sich in der Eingangsbefragung als Lernplattform-Novizen eingeschätzt hatten, ließen erkennen, dass von diesen zwar ein größerer Anteil eine längere Einarbeitungsphase benötigte (40%), mehr als die Hälfte haben sich jedoch gut bzw. nach kurzer Einarbeitung (gut 10%, kurze Einarbeitung 50%) in dem Lernraum zurecht gefunden. Das grundlegende Konzept des erprobten virtuellen Lernraumes erscheint also nach Aussage der befragten Kursteilnehmer für die Zielgruppe der Erprobung prinzipiell geeignet.

Die Auswertung der Antworten der Kursteilnehmer auf die Frage Nr. 6 zeigte bezüglich der eingesetzten Kommunikationswerkzeuge, dass die Integration von E-Mail-Funktion, Foren und Sofortmitteilungen von vielen Kursteilnehmern als sinnvoll empfunden wurde (Mehrfachantworten möglich): E-Mails bewerteten 41,7%, Foren 33,3% und Sofort-Mitteilungen 20,8% als zweckdienlich für die Kommunikation mit allen Kursteilnehmern. Für die Kommunikation in den Teilgruppen (Frage Nr. 7) fiel das Urteil der Befragten noch eindeutiger zugunsten der E-Mails aus: 54,2% der Befragten hielten dieses Instrument geeignet, um in den Teilgruppen zu kommunizieren. Die übrigen CSCL-Werkzeuge wurden höchstens von einem Viertel der Befragten als geeignet beurteilt, um mit ih-

ren Teilgruppen-Partner zu kommunizieren und kooperativ zu lernen, angeführt von den Gruppenwikis mit 25%, gefolgt von Gruppenforen (20,8%) und Sofortmitteilungen (20,8%). Schlusslicht bildeten hier die Gruppenchats (12,5%).

Diese Werkzeuge wurden darüber hinaus von zumindest einem Viertel der Novizen (laut Selbsteinschätzung) der jeweiligen Tools als geeignet beurteilt, mit der Lerngruppe im virtuellen Lernraum zu kommunizieren. Zwar scheint dies eine verhältnismäßig geringe Quote zu sein, die Novizen könnten jedoch höchstwahrscheinlich durch gezielte Übung im Umgang mit diesen Werkzeugen zu erfahreneren Benutzern werden und würden sich dann wahrscheinlich dem aus der Befragung hervorgegangenen positiven Votum der geübten Kursteilnehmer anschließen.

Bei dem CSCL-Werkzeug Wiki zeigte sich hinsichtlich der Einschätzung der Kursteilnehmer zu dessen Zweckdienlichkeit für kooperatives Lernhandeln in einem virtuellen Lernraum, dass diejenigen Befragten, die sich in der Eingangsbefragung als Neulinge dieses CSCL-Werkzeuges eingeschätzt hatten, es weitgehend als ungeeignet für kooperatives Lernen beurteilten: 27% empfanden die Gruppenwikis als nicht zweckdienlich, 32% als eher weniger zweckdienlich und nur 27% als einigermaßen zweckdienlich für die gemeinsame Bearbeitung der Gruppenaufgaben in den Teilgruppen.

Die Auswertung der freien Antworten zu Frage Nr. 9 zeigte zum Einen, dass einige Kursteilnehmer anscheinend eine generelle Abneigung gegen internetbasierte Kommunikationswerkzeuge hegten und/oder prinzipiell lieber auf für sie altbewährte Werkzeuge, wie das Telefon zurückgriffen:

„Wenn in heutigen Zeiten die junge Generation und ihr Umgang mit Chat-Räumen, Foren und die Handhabung mit ihren persönlichen Daten ständig kritisiert wird, kann ich nicht verstehen, warum ich/wir das machen sollen. (...). Ich habe das vorher nicht gebraucht und werde es auch kuenftig so oft meiden, wie es nur geht.(...)“

„Ich telefoniere lieber mit meinen Gruppenmitgliedern bzw. Kursteilnehmern.“

Am häufigsten wurden allerdings Schwierigkeiten der Kursteilnehmer bei der Kooperation untereinander geäußert, wie bspw.:

„Das Miteinander in der Gruppe klappt einfach nicht. Auf Mitteilungen wird nicht reagiert.(...)“

„(...) Ich habe einmal mündlich versucht Kontakt zu meinem Partner aufzunehmen - keine Reaktion.“

Neun von 16 Befragten, die sich zu Frage 9 geäußert hatten, machten Aussagen ähnlicher Art. Es stellte sich also heraus, dass die „Zwangseinteilung“ der Teilgruppen das Ergebnis der Evaluation stark negativ beeinflusst hat. Es wäre also ratsam gewesen, auch vor dem Hintergrund der Initiierung langfristiger Lerngruppen, den Distance-Learning-Phasen eine längerfristige Phase der Gruppenbildung voran zu stellen, inkl. einer umfangreicheren Einführung der Kursteilnehmer in die Nutzung von CSCL-Werkzeugen, als dies in dem hier beschriebenen Projekt stattgefunden hat (vgl. [Dü09], S.117 f.). Welche Auswirkungen dies wiederum auf die Nutzungsintensität des virtuellen Lernraumes gehabt hätte, da sich dann vermutlich häufig Teilnehmer zusammengetan hätten, die an der gleichen Schule arbeiten und/oder nahe beieinander wohnen und/oder arbeiten, lässt sich nicht sagen. Hierfür wäre eine Erprobung des virtuellen Lernraumes mit entsprechend geänderter Gruppeneinteilung notwendig.

4 Fazit und Ausblick

Im Rückblick auf die einleitende Fragestellung, wie das Potential von virtuellen Lernräumen in der Informatiklehrerweiterbildung genutzt werden kann, insbesondere um den

spezifischen Problemstellungen bei der Weiterbildung von praktizierenden Lehrern in Flächenländern wie Niedersachsen und Mecklenburg-Vorpommern zu begegnen, kann auf Grundlage der hier beschriebenen Untersuchung der erstellte und erprobte virtuelle Lernraum als ein geeignetes Instrument bezeichnet werden. Für allgemeingültige Aussagen bzgl. der Frage, wie ein idealer virtueller Lernraum für die Informatiklehrerweiterbildung gestaltet sein sollte, so diese denn überhaupt möglich sind, müssten weitere Überlegungen angestellt werden sowie weitere Feldversuche folgen.

Auch eine Antwort auf die Frage, ob sich mit Hilfe eines solchen virtuellen Lernraumes langfristige Lerngemeinschaften initiieren und aufrecht erhalten lassen, musste das Projekt zwangsläufig schuldig bleiben, da sie nicht als eine entsprechend langfristige Untersuchung vorgesehen war. Ob sich solche Lerngemeinschaften bilden lassen, die in einem virtuellen Lernraum über große Distanzen hinweg kollaborativ lernen und arbeiten, kann nur erhoben werden, wenn die Begleitung der Lerngruppen langfristig angelegt würde. Die Ergebnisse dieses Projekts können dafür als Vorarbeit angesehen werden. Wir halten es für durchaus denkbar, dass sich so Lerngemeinschaften bilden lassen, die auch über das Weiterbildungsstudium hinaus unter Zuhilfenahme des virtuellen Lernraumes über größere Distanzen hinweg zusammenarbeiten.

Literatur

- [Dü09] Dünnebier, M.: E-Learning in der Informatik-Lehrerweiterbildung. Masterarbeit Online unter <http://oops.uni-oldenburg.de/volltexte/2010/1117/>, Oldenburg, 2009.
- [Fu08] Fussangel, K.: Subjektive Theorien von Lehrkräften zur Kooperation. Eine Analyse der Zusammenarbeit von Lehrerinnen und Lehrern in Lerngemeinschaften. Dissertation. urn:nbn:de:hbz:468-20080475, Wuppertal, 2008.
- [FLIEG] Flexible Lehrerweiterbildung in Informatik als Erweiterungsfach für Gymnasien. <http://eddi.informatik.tu-muenchen.de/>
- [MGR95] Mandl, H.; Gruber, H.; Renkl, A.: Situiertes Lernen in multimedialen Lernumgebungen. In (Issing, L. J.; Klimsa, P. Hrsg.): Information und Lernen mit Multimedia. PVU, Weinheim, 1995; S. 167-178.
- [Mo07] Modrow, E.: Mittelstufeninformatik. Online unter: <http://www.vlin.de/vlin2/material/Mittelstufeninformatik.pdf> - zuletzt besucht 10.04.2010
- [Sc03] Schulmeister, R.: Lernplattformen für das virtuelle Lernen. Evaluation und Didaktik. Oldenbourg Wissenschaftsverlag, München, 2003.

Analyse der Studienleistungen von Studierenden an der Universität Óbuda und deren Implikationen für die Informatikausbildung

Gábor Kiss

Universität Óbuda
Budapest – Ungarn
Web: <http://uni-obuda.hu/>
E-Mail: kiss.gabor@bgk.uni-obuda.hu

Zusammenfassung: In der letzten Jahren ist die Zahl der erfolgreichen Prüfungen von Studierenden im Informatikkurs des ersten Studienjahres für verschiedene Studiengänge an der Universität Óbuda stark gesunken. Dies betrifft Prüfungen in den Teilgebieten Rechnerarchitektur, Betrieb von Peripheriegeräten, Binäre Codierung und logische Operationen, Computerviren, Computernetze und das Internet, Steganographie und Kryptographie, Betriebssysteme. Mehr als der Hälfte der Studenten konnte die Prüfungen der ersten Semester nicht erfolgreich absolvieren. Die hier vorgelegte Analyse der Studienleistungen zielt darauf ab, Gründe für diese Entwicklung zu identifizieren, die Zahl der Abbrecher zu reduzieren und die Leistungen der Studenten zu verbessern. Die Analyse zeigt, dass die Studenten die erforderlichen Lehrmaterialien erst ein bis zwei Tage vor oder sogar erst am Tag der Klausuren vom Server downloaden, so dass sie nicht mehr hinreichend Zeit zum Lernen haben. Diese Tendenz zeigt sich bei allen Teilgebieten des Studiengangs. Ein Mangel an kontinuierlicher Mitarbeit scheint einer der Gründe für ein frühes Scheitern zu sein. Ferner zeigt sich die Notwendigkeit, dass bei den Lehrangeboten in Informatik auf eine kontinuierliche Kommunikation mit den Studierenden und Rückmeldung zu aktuellen Unterrichtsinhalten zu achten ist. Dies kann durch motivierende Maßnahmen zur Teilnahme an den Übungen oder durch kleine wöchentliche schriftliche Tests geschehen.

1 Vorwort

Zunächst einige Informationen zu den organisatorischen Rahmenbedingungen der hier diskutierten Informatikausbildung. Die Universität Óbuda hat mit ihren 12000 Studierenden eine sehr gute Position in der Hochschulbildung von Budapest. Die Studenten können sich nach dem Abitur an einem Gymnasium und als Absolventen einer vierjährigen Fachschule an der Universität der Óbuda einschreiben, um ein Bachelor- oder Masterdiplom zu erwerben. Die Studenten kommen mit unterschiedlicher Vorbildung von Gymnasien und von Fachschulen an die Universität [K09]. Der Unterschied zwischen diesen beiden Schultypen besteht darin, dass im Gymnasium eher allgemein bildende Kenntnisse in Sprachen und Naturwissenschaften vermittelt werden, während in Fachschulen neben dem inhaltlich reduzierten allgemeinbildenden Fächerangebot auch fachspezifische (z.B. technische) Inhalte vermittelt werden [K10/1]. Die Einschreibung erfordert eine Mindestpunktzahl die aus den Noten der Abiturfächer ermittelt wird. Die Informatikveranstaltungen an der Bánki Donát Fakultät für Maschinenbau und Sicherheitstechnik sind nur im ersten Studienjahr obligatorisch. Die Inhalte des Informatikkurses im ersten Semester beziehen sich auf Rechnerarchitektur, Betrieb von Periphe-

riegeräten, Binäre Codierung und logische Operationen, Computerviren, Computernetze und das Internet. Inhalte des Informatikkurses im zweiten Semester sind Steganographie, Kryptographie [K10/2] und Betriebssysteme. Die Studenten, die den Informatikkurs besuchen, erhalten einen Serverzugang, über den sie seit 2007 die Lehrmaterialien für den Kurs downloaden können, um die Inhalte für sich zu Hause aufarbeiten zu können. Die Lehrmaterialien sind einen Tag nach der Vorlesung auf dem Server verfügbar. Der Server speichert, wer, wann was herunter geladen hat. Die entsprechenden Log-Dateien des letzten Studienjahres sind für diesen Artikel mit statistischen Methoden ebenfalls aufbereitet worden. Die Motivation zu dieser Analyse basiert auf der alarmierenden Einschätzung, dass gemessen an meiner achtzehnjährigen Lehrerfahrung in der Informatik sich die Abschlussergebnisse der Studenten in den letzten Jahren deutlich verschlechtert haben. Von gleichen subjektiven Erfahrungen berichten die Kollegen in den anderen Fächern der Studiengänge: Studierende scheitern zunehmend oder erzielen schlechtere Abschlussergebnisse in den Kursen. Kursergebnisse und Serverzugriffe bilden nun die Basis für die Überlegungen zum Thema in diesem Artikel.

2 Analyse der Informatik-Noten

2.1 Grundlegende organisatorische Information zu den Informatikkursen

Zunächst möchte ich beschreiben, wie die Notengebung für die Studierenden in den Informatikkursen stattfindet. Die Studenten schreiben zwei Klausuren über die Lerninhalte des Kurses in der siebten und in der letzten Woche. Die Klausuren bestehen in der Regel aus vier Fragen, die sie beantworten müssen. Jede Antwort ist in Form eines kleinen Essays von ungefähr einer halben Seite zu geben. Wenn ein Student eine Frage korrekt beantworten kann, hat er/sie die Klausur bestanden. Somit sind jeweils nur 25% der zu bearbeitenden Aufgabenstellungen für das Bestehen des Tests erforderlich. Die Studenten müssen diese zwei Klausuren insgesamt bestehen, um ein Testat für diese Veranstaltung zu bekommen. Wer dieses Testat nicht geschafft hat, hat während des Prüfungszeitraums einen zweiten Versuch. Studenten, die das Testat erworben haben, werden zu einer mündlichen Prüfung im Themengebiet zugelassen, können bei Bestehen den Kurs abschließen und werden für die Kurse des nächsten Semesters zugelassen. In Ungarn ist schlechteste Note eine Eins, die beste ist eine Fünf.

2.2 Informatiknoten

Im vergangenen Wintersemester habe ich insgesamt mehr als 700 Studenten unterrichtet. In die Analyse der Studie sind nur die Ergebnisse jener Studenten eingeflossen, die den Kurs zum ersten Mal belegt haben. Damit werden die Resultate der Studenten nicht berücksichtigt, die die Vorlesung schon einmal gehört und die Klausuren nicht bestanden haben, denn diese Ergebnisse könnten die berechneten Werte verfälschen. Die folgende Tabelle (Tabelle 1) zeigt die Anzahl bzw. den Anteil der Studierenden, sortiert nach Fachrichtung und Schultyp, die ein Testat bekommen haben, sowie jeweils den Notendurchschnitt und die Varianz.

Tabelle 1. Ergebnisse im Fach Informatik bei Fachrichtungen

Fachrichtung	Maschinenbau-Ing		Sicherheitstechnik-Ing.		Mechatronik-Ing.	
	Gymn.	Fachs.	Gymn.	Fachs.	Gymn.	Fachs.
Gesamtzahl der Studenten	158	87	59	43	29	64
ohne Testat	64,6%	58,6%	72,9%	69,8%	58,6%	57,8%
mit Testat	35,4%	41,4%	27,1%	30,2%	41,4%	42,2%
Noten-durchschnitt	2,38	2,69	2,38	2,23	2,25	2,11
Varianz der Noten	0,52	0,71	0,50	0,44	0,62	0,70

Die Werte in den Tabellen zeigen, dass mehr als die Hälfte der Studenten (bei Sicherheitstechnik Ingenieuren mehr als zwei Drittel) während des Semesters kein Testat im Lehrfach erwerben konnte. Diese Studierenden wurden demzufolge auch nicht zur Abschlussprüfung zugelassen. Die anderen, die per Testat zur Prüfung zugelassen wurden, haben in der Prüfung oft keine guten Noten erzielt. Ferner zeigt sich auch ein Unterschied zwischen den Schultypen. Im nächsten Analyseschritt sollen diese schultypspezifischen Unterschiede in den Durchschnittsnoten der Informatikabschlussprüfung genauer betrachtet werden.

2.3 Vergleich der Durchschnittsnoten nach Herkunftsschultyp

Der in den vorhergehenden Tabellen sichtbare Unterschied zwischen den Studierenden aus unterschiedlichen Schultypen in den Durchschnittsnoten wird nun einem Signifikanztest unterzogen. Mittels des Independent Sample Test können wir feststellen, ob es sich um signifikante Unterschiede handelt. Tabelle 2 zeigt die entsprechenden Ergebnisse für die Tests bei Maschinenbau-Ingenieuren.

Tabelle 2. Ergebnisse des Independent Sample Test bei Maschinenbau-Ingenieuren

	Levene's Test		T-Test		
	F	Sig.	t	Df	Sig.
	Lower	Upper	Lower	Upper	Lower
F-Test	5,975	0,016	-2,478	90,000	0,015
Welch-Test			-2,323	59,224	0,024

Die Ergebnisse des Levene's Test sind signifikant, deswegen müssen wir die Nullhypothese des F-Tests verwerfen [Link A]. Die Varianz der Noten der Studenten von verschiedenen Schultypen ist nicht gleich. Deshalb werden wir den Welch-Test benutzen, um die Durchschnittswerte zu vergleichen. Der Wert des Welch-Tests ist ebenfalls signifikant. Dies führt zu der Schlussfolgerung, dass die schulische Herkunft bei Studierenden der Fachrichtung Maschinenbauingenieuren einen relevanten Einfluss auf die erzielten Studienleistungen hat [Link B]. Die Stärke dieser signifikanten Korrelation liefert der Wert von Eta-Quadrat, wie er in Tabelle 3 zu sehen ist [K07] (Tabelle 3).

Tabelle 3. Ergebnisse der Eta-Quadrat bei Maschinenbau Ingenieuren

Informatik Note * Schultyp	Eta	Eta-Quadrat	Die Stärke dem Konnex
Maschinenbau Ingenieur	0,253	6,39%	schwach

Wir können der Tabelle entnehmen, dass die Stärke des signifikanten Zusammenhangs zwischen Schultyp und Informatiknoten bei Maschinenbau Ingenieuren nur schwach ist. Das gleiche statistische Verfahren wird nun auch auf die Daten der Studienrichtung Sicherheitstechnik-Ingenieur angewandt (Tabelle 4).

Tabelle 4. Ergebnisse des Independent Sample Test bei Sicherheitstechnik-Ingenieuren

	Levene's Test		T-Test		
	F	Sig.	t	df	Sig.
	Lower	Upper	Lower	Upper	Lower
F-Test	2,772	0,107	0,815	27,000	0,422
Welch-Test			0,827	26,810	0,416

Die Ergebnisse des Levene's Test sind nicht signifikant, deswegen können wir die Nullhypothese des F-Tests nicht verwerfen. Da die Varianz der Noten der Studenten von verschiedenen Schultypen gleich sind, müssen wir im Weiteren den T-Test benutzen, um die Durchschnittswerte zu vergleichen [Link C]. Der Wert des T-Tests ist nicht signifikant, was bedeutet, dass die Durchschnittsnoten der Studierenden der Fachrichtung ‚Sicherheitstechnik Ingenieur‘ sich hinsichtlich des Herkunftsschultyps nicht signifikant unterscheiden, diese Variable also keinen Einfluss auf die im Studium erzielten Noten hat. Betrachten wir schließlich noch die dritte Fachrichtung Mechatronik-Ingenieur (Tabelle 5).

Tabelle 5. Ergebnisse des Independent Sample Test bei Mechatronik-Ingenieuren

	Levene's Test		T-Test		
	F	Sig.	t	df	Sig.
	Lower	Upper	Lower	Upper	Lower
F-Test	0,591	0,449	-0,133	24	0,895
Welch-Test			-0,127	17,33	0,901

Die Ergebnisse des Levene's Test sind nicht signifikant, deswegen können wir die Nullhypothese des F-Tests verwerfen. Die Varianz der Noten der Studenten von verschiedenem Schultyp ist gleich. Die Anwendung des T-Tests liefert auch hier einen nicht signifikanten Wert, was bedeutet, dass die Durchschnittsnoten bei den Mechatronik-Ingenieuren ebenfalls nicht vom Herkunftsschultyp abhängen.

2.4 Download von Lehrmaterial vom Server

Die erste Klausur fand am 12.10.2009 statt, die zweite war am 30.11.2009. Die Studenten, die kein Testat erhalten haben, konnten ihren zweiten Versuch im Prüfungszeitraum am 15.12.2009 absolvieren. Anhand der Analyse der Downloaddaten können wir sehen, dass die Zahl der Downloads jeweils einen Tag vor den Klausuren am Höchsten ist. Dies

bedeutet, dass die Mehrheit der Studenten die Materialien nicht jeweils zeitnah zur Themenbehandlung in der Vorlesung bearbeitet hat, sondern erst im letzten Augenblick vor der Prüfung auf diese Materialien zugegriffen und vermutlich dann einen individuellen „Crashkurs“ absolviert hat (Abbildung 1–3). Da der Umfang der Lehrmaterialien von 5–6 Vorlesungseinheiten zu hoch ist, um erfolgreich in so kurzer Zeit bearbeitet werden zu können, ist es für die Mehrheit dieser Studenten bei diesem Arbeitsverhalten unmöglich, ein gutes Klausurergebnis zu erzielen. Die Abbildung zeigt, dass sich die Anzahl der Downloads eine Woche vor dem Klausurbeginn zu erhöhen beginnt. Ein ähnliches Muster ergibt sich auch in dem Zeitabschnitt vor der Wiederholungsprüfung.

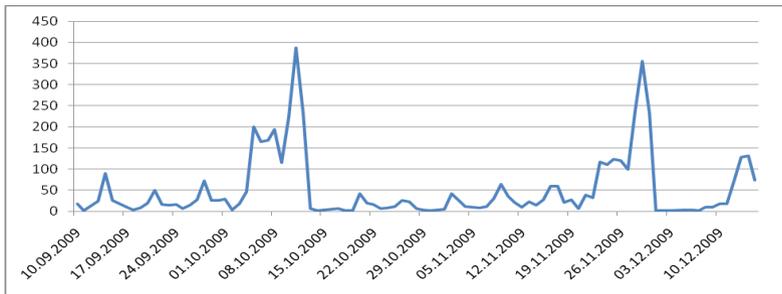


Abbildung 1. Die Zahl der Downloadzugriffe pro Tag bei den Maschinenbau-Ingenieuren

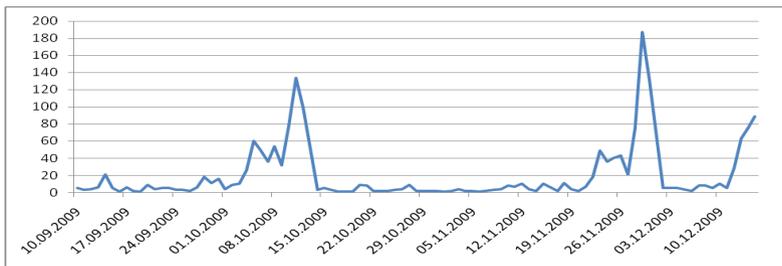


Abbildung 2. Die Zahl der Downloads pro Tag bei Sicherheitstechnik-Ingenieuren

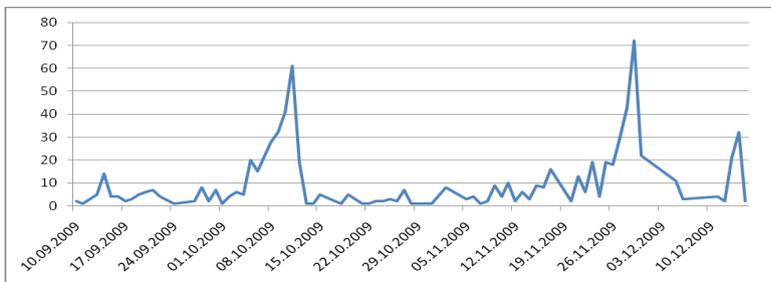


Abbildung 3. Die Zahl der Downloads pro Tag bei Mechatronik-Ingenieuren

Wir können in den drei Abbildungen jeweils die drei Spitzen in jeder der Fachrichtungen erkennen. Dies bedeutet, dass es keinen Unterschied im Arbeitsverhalten zwischen den Studenten der verschiedenen Fachrichtungen gibt. Die Mehrheit der Studenten warten bis zur letzten „Minute“, um zu lernen und sich für die Klausuren vorzubereiten. Im Sommersemester ist die Situation ähnlich. Diese typischen Downloadspitzen kann man auch dort finden. Dieses studentische Arbeitsverhalten kann einer der Gründe für die hohen Durchfallerquoten und die relativ schlechten Prüfungsergebnisse sein.

Als didaktische Idee, sich mit diesem Problem auseinanderzusetzen, wird demzufolge ein Konzept erwogen, nach dem zu Beginn jeder Vorlesungseinheit von den Studierenden schriftlich Fragen aus der letzten Vorlesungseinheit zu beantworten sind. Die Ergebnisse dieser Kurztests bilden dann eine Grundlage für die Testate und die Zulassung zur Semesterabschlussprüfung. Auf diese Weise kann der Vorlesungsbesuch zwar nicht obligatorisch gemacht, die Studenten aber zur regelmäßigen Teilnahme an der Vorlesung motiviert werden. Mit diesem Konzept soll durch den regelmäßigen Vorlesungsbesuch und die kontinuierliche Mitarbeit auch die Prüfungsleistungen der Studierenden verbessert werden.

3 Zusammenfassung

Die vorgestellte Analyse beinhaltet mehrere Ergebnisse. Erstens wir haben gesehen, dass die Studenten sich die Lernmaterialien erst in der letzten „Minute“ vor der Klausur beschaffen, um sich auf die Informatikprüfung vorzubereiten. Dies kann einer der Gründe sein, weshalb die Mehrheit der Studenten dieses Lehrfach nicht erfolgreich absolvieren, sie kein Testat bekommen und sie damit keine Prüfung in Informatik ablegen können. Ferner zeigt sich, dass die Notendurchschnittswerte ebenfalls nicht gut sind. Die Studenten nehmen sich nicht genug Zeit, die angebotenen Materialien kontinuierlich und zeitnah zur Vorlesung durchzuarbeiten. Die Analyse zeigt, dass die Lernmotivation der Studenten gering ist und sie kurz vor den Klausuren mit geringem Aufwand die notwendigen Testate erreichen wollen.

Als Konsequenz aus diesen Feststellungen ergibt sich die Notwendigkeit ein didaktisch-methodisches Konzept zu entwickeln, dass die Studierenden motiviert, sich mit den Lerninhalten kontinuierlich auseinanderzusetzen. Da es an der Universität Óbuda zur Zeit keine regelmäßige Überprüfung von Studienleistungen gibt, soll ab dem kommenden Semester vor jeder Vorlesung von den Studierenden eine Frage zu Lerninhalten der vorhergehenden Stunde beantwortet werden. Nur über die regelmäßige erfolgreiche Teilnahme an diesem Kurztest kann dann das Testat und die Prüfungszulassung erworben werden. Ferner ist über die Wiedereinführung von Übungen im ersten Semester sowie über die Einführung einer Online-Sprechstunde nachzudenken.

Ferner hat die Analyse ergeben, dass Absolventen von Fachschulen teilweise bessere fachliche Vorkenntnisse haben und entsprechend auch bessere Leistungen in Informatik erbringen. Das gilt allerdings nur für die Fachrichtung Maschinenbau wobei die Stärke des signifikanten Zusammenhangs zwischen Schultyp und Informatiknoten bei Maschinenbau Ingenieuren nur schwach ausgeprägt ist. Als Konsequenz dieser Analyse ergibt sich, dass Absolventen von Gymnasien nur teilweise einer besonderen Betreuung und Förderung in Informatik bedürfen.

Literatur

- [K07] Korpás Attiláné dr.: Általános statisztika I. 152-153. old.
- [K09] Gabor Kiss – The survey measuring the informatics skills of the entering Students at Budapest Tech, Bánki Donát Faculty of Mechanical and Safety Engineering / 7th IEEE International Symposium on Intelligent System and Informatics, Subotica, Serbia, ISBN: 978-1-4244-5348-1, pp:395-397, 2009, Digital Object Identifier: 10.1109/SISY.2009.5291125
- [K10/1] Gábor Kiss- Measuring Computer Science knowledge at the end of secondary grammar school in Hungary (pp: 839-842) / International Educational Technology Conference (IETC 2010), 2010, Istanbul
- [K10/2] Gabor Kiss - Experiences in teaching data concealment and data encryption to engineering undergraduates / 9th IEEE International Conference on Information Technology Based Higher Education and Training (ITHET 2010), Cappadokia, ISBN 978-1-4244-4811-1, pp:419-423, IEEE Catalog Number: CFP10587-CDR, Digital Object Identifier: 10.1109/ITHET.2010.5480011, Indexed by EI Compendex
- [Link A] http://en.wikipedia.org/wiki/Levene's_test
- [Link B] http://en.wikipedia.org/wiki/Welch's_t_test
- [Link C] http://en.wikipedia.org/wiki/Student's_t-test

Informatik für Migrantinnen und Migranten Einführung eines neuen Studienprogramms an der Universität Oldenburg

Susanne Boll, Rolf Meinhardt, Sabine Gronewold, Larissa Krekeler

Fakultät II Department für Informatik – Carl-von-Ossietzky Universität Oldenburg
26111 Oldenburg - Germany

Web: www.informatik.uni-oldenburg.de/studium/informatikfuermigranten

E-Mail: informatik-fuer-migranten@informatik.uni-oldenburg.de

Zusammenfassung: Für die Integration und den Bedarf der hochqualifizierten Migranten auf dem Arbeitsmarkt in Deutschland gibt es viele Überlegungen, aber noch keine ausreichenden Lösungen. Dieser Artikel beschreibt eine praktische Lösung über die Umsetzung des Konzepts für die Qualifizierung der akademischen Migranten am Beispiel eines Studienprogramms in Informatik an der Universität Oldenburg.

Warum ein Studienprogramm Informatik für Migranten?

Nach Recherchen des Instituts der deutschen Wirtschaft Köln (IW) fehlen in Deutschland 65.000 MINT-Fachkräfte (Mathematiker, Informatiker, Naturwissenschaftler und Techniker). Dabei sind insgesamt 2,8 Millionen Migranten fertig ausgebildet nach Deutschland gekommen: Die Akademiker unter ihnen hatten zu etwa 60 % ihr Studium im Herkunftsland absolviert, die übrigen kamen zu 44 % mit einem Berufsabschluss nach Deutschland (vgl. [I09]). Allerdings gibt es deutliche Schwierigkeiten bei der Anerkennung der im Ausland erworbenen Abschlüsse und damit einhergehend Probleme, in den Arbeitsmarkt einzutreten. Zentrale Ursache ist die geringe Anerkennung der Abschlüsse: etwa 65 % der Anträge werden abgelehnt. Obwohl keine flächendeckenden Studien vorliegen, lassen einzelne Erhebungen einen Schluss auf einen echten Bedarf an der Integration hochqualifizierter Migranten zu. Etwa 5–10% der Aussiedler und knapp 20% der Flüchtlinge haben einen akademischen Abschluss; bei jüdischen Immigranten aus der ehemaligen Sowjetunion liegt dieser Anteil sogar bei 70–80% (vgl. [M08]). Nach Schätzungen der Universität Oldenburg lebten insgesamt eine halbe Million ausländischer Fachkräfte in Deutschland, deren Abschlüsse größtenteils nicht anerkannt sind. Auch Staatsministerin Maria Böhmer (Beauftragte der Bundesregierung für Migration, Flüchtlinge und Integration) beziffert im November 2009 das Potential mit „mehreren hunderttausend qualifizierten Zuwanderern aus verschiedensten Berufen“ (siehe auch [P09]). Allerdings bleibt auch nach Anerkennung der Abschlüsse eine Hürde bei der Integration in den Arbeitsmarkt: Arbeitgeber können die Qualifikation der ausländischen Abschlüsse, die teilweise auch schon einige Zeit zurückliegen, oft nicht einschätzen. Bisher liegen zwar kaum gesicherte Erkenntnisse darüber vor, welche Qualifikationsprofile höher qualifizierte Migranten in Deutschland aufweisen und in welchem Maße ihre mitgebrachten Qualifikationen anerkannt werden. Dennoch zeigen Studien,

dass Verfahren zur Anerkennung und ressourcenorientierte Qualifizierungsmaßnahmen zu einer verbesserten beruflichen und sozialen Integration führen. Der im Jahr 2006 an der Universität Oldenburg eingerichtete Weiterbildungsstudiengang „Interkulturelle Bildung und Beratung“ mit dem Abschluss „Bachelor of Arts“ zeigte bereits, dass es besonders motivierend für die Absolventen ist, dass sie einen „normalen“, in ganz Europa anerkannten berufsbefähigenden Abschluss erhalten (vgl. [M10]). Viele Migranten haben zuvor bereits Zertifikate im Inland und Ausland, aber keinen formalen oder anerkannten Abschluss erworben. Damit entsteht die besondere Herausforderung der Integration von hochqualifizierten Migranten unter Berücksichtigung der bereits erworbenen Qualifikationen als auch die Anforderungen des deutschen Arbeitsmarktes. Da zum einen eine vollständige Anerkennung und Integration in den Arbeitsmarkt nicht gelingt, zum anderen aber eine Vielzahl von relevanten Vorkenntnissen vorhanden ist, stellt sich die Frage, wie ein Studienkonzept in der Informatik aussehen kann, das die Migranten aus ihrer jeweiligen Bildungs- und Lebensbiographie abholt und ihnen einen zügigen Weg in den Arbeitsmarkt bietet.

Aus diesem Grund wurde im Jahr 2008 an der Carl-von-Ossietzky Universität, unterstützt vom Niedersächsischen Ministerium für Inneres, Sport und Integration, das Studienprogramm „Informatik für Migrantinnen und Migranten“ entwickelt. Das Studienprogramm berücksichtigt die besondere Situation der Migranten. Diese haben oft zwar ein sehr gutes fachliches Fundament im Bereich der theoretischen, formalen und mathematischen Methoden, es fehlen jedoch aktuelle, praxisorientierte Themen der Informatik wie Programmiersprachen oder modernes Software-Engineering, um auf dem deutschen Arbeitsmarkt trotz des derzeit herrschenden Fachkräftemangels die passende Qualifikation aufweisen zu können. Da wie bereits erwähnt eine vollumfängliche Anerkennung eines vorangehenden Studiums oft nicht gelingt, prüft das Studienprogramm individuelle mögliche Anerkennungen, um eine gute Balance zwischen dem bestehenden Vorwissen und den wettbewerbsfähigen neu zu erwerbenden oder aufzufrischenden Kenntnissen und Fertigkeiten in Informatik für den deutschen Arbeitsmarkt herzustellen.

Das Studienprogramm „Informatik für Migranten“

Das Studienprogramm „Informatik für Migrantinnen und Migranten“ wurde im Wintersemester 2008/2009 für hochqualifizierte Migranten, die bereits in ihrem Heimatland ein Studium absolviert haben, eingerichtet, um ihnen – über ein weiterqualifizierendes Studium – einen in Deutschland anerkannten und berufsfähigen Abschluss zu ermöglichen (siehe auch [S10]). Die Teilnehmenden an dem Studienprogramm absolvieren das Studium in einem der Studiengänge der Informatik:

- Fach-Bachelor Informatik,
- Fach-Bachelor Wirtschaftsinformatik,
- Zwei-Fächer Bachelor Informatik,
- Fach-Master Informatik,
- Fach-Master Wirtschaftsinformatik,
- Fach-Master Eingebettete Systeme und Mikrorobotik,
- Master of Education mit Fach Informatik.

Die Art des Studienganges richtet sich zum einen nach der Qualifikation, zum anderen aber auch an dem Interesse aus, das die Studierenden mitbringen. Studierende im Studi-

enprogramm sind reguläre Studierende und Mitglieder der Universität Oldenburg und haben als Studierende auch die anfallenden Semester- und Studienbeträge zu übernehmen. Das Studium, sowohl in den Bachelor- als auch in den Masterstudiengängen kann als Vollzeit- oder als Teilzeitstudium durchgeführt werden.

Vorhandene akademische und berufliche Leistungen werden nach Möglichkeit für das Bachelor- bzw. Masterstudium angerechnet und führen so zu einer Verkürzung der Studienzeit. Fehlende bzw. veraltete Kompetenzen werden über das Studium aufgefrischt bzw. neu erworben. Die Besonderheit des Studienangebotes ist also nicht etwa ein speziell auf die Anforderungen von gut ausgebildeten Migranten gerichtetes Profil, sondern die individuell zugeschnittene Studienplanung und -begleitung. So können die Migranten ihre bisherige Ausbildung gezielt verwenden und ihr Potenzial zielgerichtet ausschöpfen. Sie werden dabei durch intensive Betreuung und individuelle Beratung und Studienplanung unterstützt. Für den Aufbau und die Einführung des Studienprogramms „Informatik für Migrantinnen und Migranten“ war die Konzeption und Umsetzung einer Reihe von Bausteinen notwendig, die im Folgenden kurz vorgestellt werden.

Bewerbung und Aufnahme

Voraussetzung für eine Bewerbung um einen Platz im Studienprogramm sind eine Zulassung zum Universitätsstudium in Deutschland sowie ein ausländischer Hochschulabschluss. Die Teilnehmer durchlaufen als angehende Studierende der Universität Oldenburg den „regulären“ Weg bis zur Immatrikulation: Sie müssen das für ein Studium an einer deutschen Hochschule notwendige Sprachniveau erreicht haben und es muss eine positive Prüfung der Bewerbungsunterlagen über die Arbeits- und Servicestelle für Internationale Studienbewerbungen e.V. (uni-assist) in Berlin erfolgen. Eine Betreuung der Teilnehmer durch das Studienprogramm findet in der Regel schon in der Bewerbungsphase um einen Studienplatz statt. Durch eine enge Zusammenarbeit mit dem Immatrikulationsamt und dem Sprachenzentrum der Universität konnten die Abläufe optimal gestaltet werden.

Anerkennung und Anrechnung

Zentrale Motivation für Migranten ist es, einen anerkannten Studienabschluss in Deutschland zu erlangen und gleichzeitig die bisherigen Fachkenntnisse in das Studium einzubringen. Dabei kann es notwendig sein, Kenntnisse entsprechend aufzufrischen und zu ergänzen. Die Prüfungsordnungen der Universität Oldenburg sehen Regelungen vor, nach denen Leistungen, die an ausländischen Hochschulen erbracht wurden, anerkannt werden. Darüber hinaus erlaubt die Prüfungsordnung die Anerkennung von „außerhalb des Studiums abgeleisteten Prüfungsleistungen“. Die konkrete Anrechnung der Leistungen erfolgt über den Prüfungsausschuss Informatik. In der aufwändigen Vorbereitung der Anrechnung und Überprüfung der Gleichwertigkeit wird der Prüfungsausschuss durch die Mitarbeiterinnen im Studienprogramm unterstützt. Der Katalog der anzurechnenden Vorkenntnisse wird dem Prüfungsausschuss Informatik vorgelegt, der die Anrechnung durchführt. Die Studierenden schreiben sich zunächst für das erste Fachsemester der Bachelor- bzw. Master-Studiengänge ein und werden nach einer Anrechnung in das entsprechende Fachsemester hochgestuft.

Entwicklung eines individuellen Studienplanes

Durch die modulare Organisation¹ des Studiums und durch die Vergabe von Kreditpunkten als Maßeinheit für den Arbeitsaufwand ist das Studium der Informatik in Oldenburg klar strukturiert (siehe Beispielstundenplan in Abschnitt 3.1, Abbildung 1 und 2). Eine individuelle Planung des Studiums ist zwar möglich, erfordert aber die sorgfältige Beachtung aller Gegebenheiten, um das Studium organisatorisch und damit auch zeitlich optimal zu gestalten. Durch die Anrechnung der Leistungen aus dem vorherigen Studium ist es notwendig, die Studienpläne individuell zu gestalten und auf die jeweiligen Bedürfnisse anzupassen. Nach ausführlicher Prüfung der Unterlagen wird ein Studienplan in Kooperation mit den Studierenden entwickelt, der die fehlenden Module bis zum angestrebten Abschluss zusammenstellt und einen sinnvollen Studienverlauf beschreibt. Dieser Plan wird im weiteren Studium von Semester zu Semester aktualisiert und auf die Bedürfnisse und Interessen der Studierenden angepasst.

Ergänzende Veranstaltungen

Zusätzlich zur Anerkennung von Vorkenntnissen werden für das Studienprogramm ergänzende Fachkurse entwickelt und angeboten. Diese dienen dazu, die fachlichen Kompetenzen der Studierenden zu stärken und anerkannte Vorkenntnisse in den verschiedenen Grundlagen der Informatik aufzufrischen und zu aktualisieren. Die Kurse werden nach Bedarf der Gruppe angeboten. Das Angebot umfasst dabei insbesondere Bereiche der technischen Grundlagen der Informatik und Programmierung, der Fachsprache sowie auch speziell für Lehramtsstudierende Vorbereitungskurse auf den Schuldienst.

Beratung und Begleitung der Studierenden

Die Betreuung im Rahmen des Studienprogramms sieht eine intensive Beratungsphase vor Studienbeginn sowie Begleitung im weiteren Studienverlauf vor. Wie oben beschrieben erfolgt zunächst eine individuelle fachliche Beratung verbunden mit der Anrechnung der Leistungen und der Studienverlaufsplanung. Zudem werden zum Studienbeginn sämtliche organisatorische Fragen geklärt und grundlegende Informationen zum Studium an der Universität Oldenburg vermittelt. Die Studierenden können außerdem an allen studieneinführenden Veranstaltungen teilnehmen, die im Department für Informatik für die Studienanfänger angeboten werden: Zu Beginn des Wintersemesters (eine Woche vor Vorlesungsbeginn) finden im Rahmen der Orientierungswoche Veranstaltungen für alle Studierenden der Informatik und Wirtschaftsinformatik statt, in der vielfältige Informationen zum Studium und zur Studienorganisation vermittelt werden. Hier bieten sich auch gute Möglichkeiten, andere Studierende sowie Lehrende direkt zu Studienbeginn kennenzulernen. Seit einigen Jahren werden ebenfalls im Wintersemester Sozialtutorien („Ersti-Tutorien“) angeboten, um die Studienanfänger in der Eingewöhnungsphase im Studium zu unterstützen. Ein wichtiger Aspekt dieser Tutorien ist das Kennenlernen von Kommilitonen, was nicht selten in die Bildung studienbezogener Arbeitsgruppen mündet. Eine Teilnahme an diesen Tutorien ist für die Studierenden des Studienprogramms zur Integration in das Studium sehr hilfreich. Nach dieser Einführungsphase werden die Studierenden im Rahmen des Studienprogramms über regelmäßige wöchentliche Treffen

¹ Ein Modul ist eine Lehreinheit, die in den Studiengängen der Informatik in der Regel vier Semesterwochenstunden (SWS) und damit sechs Kreditpunkte umfasst.

weiterbegleitet. Diese Treffen bieten die Möglichkeit, Fragen zum Studium zeitnah zu klären, bei auftretenden Problemen zu beraten und zu vermitteln, Begleitmaterialien zum Studium bereitzustellen, den Studienfortschritt zu überwachen und die Studierenden zu motivieren.

Das Studienprogramm im Betrieb – erste Erfahrungen

Das Studienprogramm „Informatik für Migrantinnen und Migranten“ ist im Wintersemester 2008/2009 mit vier Studierenden gestartet. In den letzten zwei Semestern wurden insgesamt 16 weitere Teilnehmer aufgenommen, so dass aktuell elf Studierende im Vorsemester „Deutsch“, ein Studierender im Bachelor-Studiengang Wirtschaftsinformatik, drei Studierende im Bachelor-Studiengang Informatik, eine Studierende im Zwei-Fächer-Bachelor Informatik und eine Studierende im Master-Studiengang Wirtschaftsinformatik studieren. Drei Studierende haben in der Zwischenzeit ihr Studium wieder abgebrochen.

Die Teilnehmer stammen aus Russland und der Ukraine, aus dem Irak, der Türkei und Argentinien. Das Alter liegt zwischen 28 und 46 Jahren. Die Studierenden sind zu einem großen Teil verheiratet und haben Kinder. Zwei der Teilnehmer wohnen direkt in Oldenburg, die Wohnorte der anderen Studierenden liegen bis zu 100 km von Oldenburg entfernt.

Bisher gab es weitaus mehr Interessierte und aussichtsreiche Bewerbungen für das Studienprogramm, als es die Zahl der derzeitigen Studierenden widerspiegelt. Fehlende fachliche und sprachliche Kompetenzen waren die Hauptgründe für Absagen seitens des Studienprogramms. Absagen seitens der Bewerber dagegen resultierten überwiegend aus finanziellen Problemen wegen der Studiengebühren und der Lebenshaltungskosten. Hier fehlen dringend Finanzierungsmöglichkeiten wie BAföG o.Ä.

Individuelle Studienplanung – zwei exemplarische Beispiele

Im Folgenden werden beispielhaft die Studienverlaufspläne zweier Teilnehmer des Studienprogrammes dargestellt und beschrieben. Abbildung 1 zeigt den Studienplan einer Studierenden des Studienprogramms im Fachbachelor Informatik. Sie hat ihr Studium zum Wintersemester 2008/2009 mit der Auflage, die DSH-Prüfung innerhalb von zwei Semestern studienbegleitend nachzuholen, aufgenommen und bringt ein fachfremdes Ingenieurstudium in ihr aktuelles Studium ein. Ihre Studienleistungen aus dem Erststudium im Heimatland lassen sich im Rahmen von neun Modulen (drei Mathematik-Module und sechs Nebenfach-Module) auf das Studium anrechnen. Aufgrund des belegten Sprachkurses und aufgrund von fachlichen Problemen wurden im ersten Semester keine Informatikmodule bestanden. Im Sommersemester 2009 absolvierte sie zwei Module, im Wintersemester 2009/10 fünf Module und im Sommersemester 2010 vier Module. Es fehlen aktuell noch sechs Module aus dem Bereich Informatik, ein Praxismodul sowie die Abschlussarbeit, damit könnte das Studium voraussichtlich zum Ende des Sommersemesters 2011 beendet werden.

Erstes Jahr Semester 1	Algorithmen und Datenstrukturen 1 (WS 2009/10)	Programmierkurs (WS 2009/10)	Technische Informatik 1 (WS 2009/10)	Diskrete Strukturen (WS 2010/11)	Mathematik für Informatik 1 (Anrechnung)
Erstes Jahr Semester 2	Algorithmen und Datenstrukturen 2 (SS 2009)	Soft Skills (SS 2010)	Technische Informatik 2 (SS 2010)	Theoretische Informatik 1 (SS 2009)	Mathematik für Informatik 2 (Anrechnung)
Zweites Jahr Semester 3	Software Engineering (WS 2010/11)	Praktische Informatik (WS 2009/10)	Wahl 1 Petrinetze I (SS 2010)	Theoretische Informatik 2 (WS 2010/11)	Mathematik speziell (Anrechnung)
Zweites Jahr Semester 4	Softwareprojekt inklusive Proseminar (SS 2010 & WS 2010/11)	Wahl 2 Rechnernetze (SS 2011)	Praktikum Technische Informatik (SS 2011)	Wahl 4 (Anrechnung Nebenfach)	Wahl 5 (Anrechnung Nebenfach)
Drittes Jahr Semester 5		Informatik und Gesellschaft (WS 2010/11)	Wahl 3 Medienverarbeitung (SS 2011)	Wahl 6 (Anrechnung Nebenfach)	Wahl 7 (Anrechnung Nebenfach)
Drittes Jahr Semester 6	Individuelles Projekt inklusive Präsentation und Abschlussarbeit (SS 2011)			Wahl 8 (Anrechnung Nebenfach)	Wahl 9 (Anrechnung Nebenfach)

Abb. 1: Individueller Studienplan für eine Studierende im Studiengang Fachbachelor Informatik

Erstes Jahr Semester 1	AM1 ² Software Engineering (WS 2008/09)	AM 2 Wirtschaftsinformatik II (SS 2009)	AM 3 E-Commerce (E-Learning) (SS 2009)	BW 1 ³ ERP-Technologien (SS 2009)	BW2 ⁴ Software System Engineering (SS 2009)
Erstes Jahr Semester 2	Projektgruppe inklusive Seminarvortrag und Abschlussbericht idR. mit Themenbezug zur Wirtschaftsinformatik (WS 2009/10 & SS 2010)		BW1 Customizing (Block) (SS 2009)	BW2 Betriebliche Informationssysteme (SS 2009)	BW3 Unternehmensstrategien (WS 2009/10)
Zweites Jahr Semester 3			BW 1 IT-Controlling (WS 2009/10)	BW2 Software Usability Engineering (WS 2009/10)	BW3 ⁵ Internationales Management (SS 2010)
Zweites Jahr Semester 4	Master-Arbeit inklusive Präsentation idR. mit Themenbezug zur Wirtschaftsinformatik (WS 2010/11)				

Abb. 2: Individueller Studienplan für eine Studierende im Studiengang Master Wirtschaftsinformatik

In Abbildung 2 wird der Studienverlaufsplan einer Studierenden des Studienprogramms im Master Wirtschaftsinformatik dargestellt. Sie hat ihr Studium ebenfalls zum Wintersemester 2008/2009 aufgenommen, in Russland ein Diplom in Wirtschaftswissenschaften erworben und in Deutschland eine Ausbildung im kaufmännischen sowie einige

2 AM = Angleichmodul. Angleichmodule werden vom zuständigen Zulassungsausschuss individuell festgelegt.

3 BW1 = Module aus dem der Bereich Wirtschaftsinformatik.

4 BW2 = Module aus dem Bereich der Angewandten oder Praktischen Informatik.

5 BW3 = Module aus dem Bereich Betriebswirtschaftslehre.

Fortbildungen im IT-Bereich absolviert. Ihre Studienleistungen aus dem Erststudium im Heimatland wurden wie ein Bachelorstudium angerechnet, so dass sie direkt in das Masterstudium einsteigen konnte. Sie musste in ihrem Studium insgesamt elf Module, eine Projektgruppe sowie die Master-Abschlussarbeit absolvieren. Sie hatte zu Beginn des Studiums einige Eingewöhnungsschwierigkeiten, so dass nur ein Modul bestanden wurde. Inzwischen hat sie alle Module bis auf die Master-Abschlussarbeit erfolgreich absolviert, so dass sie ihr Studium voraussichtlich gegen Ende des WS 2010/2011 beenden wird.

Die beiden Studienpläne zeigen unterschiedliche Studienverläufe, abhängig von der mitgebrachten Qualifikation. Während die Studierende im Fachbachelor Informatik zwar mit ingenieurwissenschaftlichem Background, aber bezogen auf die Informatik fachfremd ins Studium eingestiegen ist, hat die Studierende im Master Wirtschaftsinformatik aufbauend auf ihr Erststudium in relativ kurzer Zeit eine Art Weiterbildungsstudium absolviert. Beide werden in Kürze ihr Studium abschließen und dem Arbeitsmarkt als qualifizierte Fachkräfte zur Verfügung stehen.

Beratung und Begleitung – Erfahrungen im ersten Durchgang

In der Startphase des Projekts wurde festgestellt, dass der Beratungsbedarf für die Gruppe der Studierenden mit Migrationshintergrund sehr hoch ist. Die Studierenden waren unerfahren und unsicher im deutschen Studiensystem und hatten einen großen Unterstützungsbedarf. Die Integration der Studierenden in die Studiengänge der Informatik konnte daher nur sehr langsam stattfinden: Die Teilnehmer orientierten sich nur schwer und fanden daher schlecht Anschluss an die Gruppe der übrigen Studierenden. Hinzu kamen sprachliche Schwierigkeiten: Durch fehlende Deutsch-Kenntnisse konnten Texte (insbesondere Aufgaben und Klausuren) nur mit Mühe und großem Zeitaufwand bearbeitet werden. Auch das geringe Sprachverständnis für die Fachsprache verursachte Probleme. Eine intensive Begleitung der Studierenden insbesondere in der Anfangsphase des Studiums erwies sich daher als unbedingt notwendig.

Ein Netz von Maßnahmen wurde zu diesem Zweck installiert: Als Erste-Hilfe-Maßnahme wurde ein „Nachhilfetutor“ eingestellt, der die Studierenden fachlich individuell im ersten Semester begleitete. Außerdem wurden die Studierenden motiviert, regelmäßig an den Ersti-Tutorien teilzunehmen. Das Angebot der Deutsch-Kurse wurde speziell im Basisbereich verstärkt. Zudem fand eine umfangreiche Betreuung der Studierenden durch das Studienprogramm in Form einer Begleitung und allgemeinen Beratung rund um das Studium statt. Nach einer Eingewöhnungszeit wurden dann zusätzliche Ergänzungskurse im Bereich Informatik und Mathematik angeboten. Zum Ende des Sommersemesters 2009 fand ein Propädeutikum zum Modul „Programmierkurs Java“ statt und in den Semesterferien nach dem Wintersemester 2009/10 wurde ein „Auffrischkurs Mathematik“ angeboten. Diese Kurse hatten, neben der Auffrischung von Fachkenntnissen und dem „Vertraut machen“ mit den deutschen Fachbegriffen, zudem den Vorteil, den Arbeitsaufwand für die entsprechenden Fachmodule, die im darauffolgenden Semester belegt werden mussten, etwas zu verringern. Auf diese Weise war der Arbeitsaufwand insgesamt für die Studierenden besser zu bewältigen.

Neben den fachbezogenen und sprachlichen Schwierigkeiten traten Probleme bezüglich der Finanzierung des Studiums auf: Allein für das Studium fallen Kosten in Höhe von 780€ im Semester an. Hinzukommen Material- und Fahrtkosten sowie Kosten für den Lebensunterhalt. Viele Studierende hatten bisher Unterstützung durch das Arbeitslosen-

geld 2 (ALG 2)⁶, zumindest ergänzend, erhalten. Diese fiel mit Studienbeginn weg; ein Bafög-Anspruch bestand aber aufgrund des Alters bzw. des vorhergehenden Abschlusses meist nicht. Zur Sicherung des Familieneinkommens musste einer bezahlten Tätigkeit nachgegangen werden, wodurch das Studium erschwert und zeitlich verlängert wurde. Inzwischen haben sich die Probleme relativiert: die Studierenden haben sich in Lerngruppen organisiert oder Anschluss an bestehende Lerngruppen gefunden, so dass die Arbeit für die einzelnen Module gut zu bewältigen ist. Die sprachlichen Probleme werden langsam kompensiert. Die Studierenden sind zwar meist mit nur zwei bis drei belegten Modulen ins Studium eingestiegen, aber schon im dritten Semester absolvieren sie mehrheitlich den regulären Stundenplan mit bis zu fünf Modulen.

Fazit

Mit dem Studienprogramm an der Universität Oldenburg wurde bundesweit ein erstes Studienangebot für hochqualifizierte Migranten im Bereich der Informatik angeboten. Das Konzept der Begleitung und Beratung hat sich als dringend notwendig und sehr erfolgreich erwiesen. Noch stärker als zunächst schon angenommen, hat die erste Phase des Studienprogramms gezeigt, wie wichtig die Integration in die Lehr-, Lern- und Studienstrukturen ist. Das mit dem Studienprogramm etablierte Begleitprogramm von der Auswahl über die individuelle Beratung und Begleitung ist neben der Fachausbildung eine wichtige Säule der Informatikausbildung für diese besondere Zielgruppe.

Das Interesse am Studienprogramm ist groß – trotz bisher noch eher geringer Werbung und Bekanntheit des Programms. Aufgrund von zwei Interviews, die in den letzten zwei Monaten veröffentlicht wurden, gab es deutlich mehr Anfragen aus dem gesamten Bundesgebiet. Es resultierten daraus 22 konkrete Bewerbungen, von denen elf direkt in Studienprogramm aufgenommen wurden. Mit den neu aufgenommenen Bewerbern ist ein altes Problem wieder in den Fokus der Arbeit gerückt – die Finanzierung des Studiums. Es gibt für diese Zielgruppe keine ausreichenden Lösungen und daher muss dringend nach Möglichkeiten der Abhilfe gesucht werden.

Literatur

- [I09] Institut der deutschen Wirtschaft Köln: Informationsdienst der deutschen Wirtschaft, Ausgabe vom 9.07.2009, <http://www-ivkoeln.de/Publikationen/iwd/Archiv/tabid/122/articleid/23673/Default.aspx>
- [M08] Meinhardt, Rolf: Die ignorierte Elite – zur prekären Lage hochqualifizierter Einwanderer und der Entwicklung von Studienangeboten zu ihrer beruflichen Integration, in Lange, Dirk (Hrsg.): Migration und Bürgerbewusstsein, Wiesbaden 2008
- [M10] Meinhardt, Rolf: Akademische Weiterbildung für hochqualifizierte Einwanderer: ein Win-Win-Angebot, in Koch, Eckart, Speiser, Sabine (Hrsg.), Internationale Migration: Chancen und interkulturelle Herausforderungen, München, Mering 2010
- [P09] Presse- und Informationsamt der Bundesregierung: Integrationspolitik: Gleichberechtigte Teilhabe – der demografische Wandel bringt auch Chancen, Pressemitteilung 493 zum Kongress „Demografischer Wandel: Integration und Gesundheit“, Hannover 18.12.2009, <http://www.bundesregierung.de/Content/DE/Pressemitteilungen/BPA/2009/12/2009-12-18-ib-demografischer-wandel.html>
- [S10] Carl-Ossietsky Universität / Department für Informatik: Studienprogramm Informatik für Migrantinnen und Migranten, Oldenburg 2010, <http://www.informatik.uni-oldenburg.de/studium/informatikfuermigranten>

⁶ Arbeitslosengeld 2 (ALG 2): Grundsicherungsleistung für erwerbsfähige Hilfebedürftige nach dem Sozialgesetzbuch II.

Was ist eine kreative Leistung Studierender? Erfahrungen eines kreativitätsförderlichen Lehrbeispiels

Isa Jahnke, Tobias Haertel, Volker Mattick, Karsten Lettow

TU Dortmund
Hochschuldidaktisches Zentrum & Fakultät für Informatik
www.hdz.tu-dortmund.de

E-Mail: [isa.jahnke; tobias.haertel; volker.mattick; karsten.lettow]@tu-dortmund.de

Zusammenfassung: Was ist eine kreative Leistung von Studierenden? Dies ist die Ausgangsfrage, wenn Lehre kreativitätsförderlicher als bislang gestaltet werden soll. In diesem Beitrag wird ein Modell zur Förderung von Kreativität in der Hochschullehre vorgestellt und mit einem Beispiel verdeutlicht. Es wird die veränderte Konzeption der Vorlesung Informatik & Gesellschaft illustriert: Studierende hatten die Aufgabe, eine „e-Infrastruktur für die Universität NeuDoBoDu“ zu entwickeln. Hierzu werden die Ergebnisse der Evaluation und Erfahrungen erläutert.

1 Einleitung: Kreative Leistungen Studierender

Anderson und Krathwohl ([AK01]) re-organisierten in Anlehnung an Benjamin Bloom's Taxonomie ([B156]) die Stufen des Kompetenzerwerbs neu: 1. Remembering, 2. Understanding, 3. Applying 4. Analyzing 5. Evaluating und 6. Creating. Oftmals geht es in der Hochschullehre nicht über die ersten zwei Stufen „Wissen reproduzieren“ und „Verstehen“ hinaus. Insbesondere die letzte Stufe „creating“ wird in Universitäten weniger stark fokussiert. Dieser Artikel setzt an Stufe 6 an. Es werden, empirisch-basiert, hochschuldidaktische Gestaltungsansätze für die Stufe „creating“ geliefert. Diese basieren auf Daten (Interviews, Online-Befragungen in verschiedenen Disziplinen mit Lehrenden) aus dem DaVinci-Projekt (gefördert vom BMBF).

Kreativität ist ein facettenreicher Begriff. Er wird oft in Zusammenhang gebracht mit KünstlerInnen, MusikerInnen, ErfinderInnen, SchriftstellerInnen und immer öfter auch mit Studierenden. Auf dramatische Weise macht Wolf Wagner [Wa10] in seinem Buch „Tatort Universität“ auf den Zusammenhang einer Kreativität fördernden Hochschulausbildung und einer innovationsfreudigen, zukunftsfähigen Gesellschaft aufmerksam und bemängelt, dass „Deutschland [...] im Hinblick auf Innovationsfähigkeit hinter allen vergleichbaren Industrieländern zurück [bleibt].“ (ebd., S. 15). Um das „stille Drama“ (ebd.) an deutschen Universitäten zu beenden und das kreative Potenzial der Studierenden stärker zu fördern, schlägt Wagner eine Reihe von Maßnahmen vor, die überwiegend an der Institution Hochschule und den Curricula ansetzen. In diesem Beitrag wird aus hochschuldidaktischer Sicht, im Bereich konkreter Lehr-/Lernszenarien, gezeigt werden, wie Kreativität von Studierenden gefördert werden kann, und wie neue Web-2.0-Techniken dazu genutzt werden können, dieses Ziel zu erreichen.

Dazu muss zunächst geklärt werden, was genau es zu fördern gilt, wenn im Kontext von Universitäten von „Kreativität“ gesprochen wird. In der wissenschaftlichen Literatur fin-

den sich vielfältige Definitionen und Konzepte von Kreativität, die sich teilweise widersprechen und gegenseitig ausschließen (s. z.B. [So07]). Allgemein lässt sich festhalten, dass etwas kreativ ist, wenn es neu und wertvoll bzw. nützlich ist ([Br06], [St99]). Jedoch stellt sich die Frage, wer bzw. welche Personen entscheiden, wann etwas als neu, wertvoll oder nützlich zu bewerten ist: In den allgemeinen Definitionen wird die Güte bzw. der Grad der Kreativität neuer Ideen bisweilen daran bemessen, inwiefern andere ihr Kreativität zuschreiben. Kreativität (Ideengenerierung) und Innovation (Ideenakzeptanz/ -durchsetzung) werden bei diesen Ansätzen nicht genügend getrennt. Es ist fraglich, ob die Sichtweise – andere bewerten was kreativ ist oder nicht ist – auch für die Hochschullehre geeignet und angemessen ist. Ziel in der Lehre ist es, Studierenden zu ermöglichen, vielfältige Kompetenzen zu erwerben. Bezogen auf Kreativität bedeutet dies, die Entfaltung ihrer eigenen Kreativität zu unterstützen und sie darin zu befähigen, sowohl in ihren Lernprozessen, als auch in ihren zukünftigen beruflichen Kontexten *kreativ* Lösungen erarbeiten zu können. Dazu gehört auch, neue, unkonventionelle Ideen zu entwickeln und sich dabei nicht in jedem Fall durch die Beurteilung von Dritten beeinflussen zu lassen.

2 Empirische Ergebnisse: ein 6-Stufen-Modell

Um für den speziellen Kontext der Hochschullehre den Begriff „Kreativität“ exakter eingrenzen zu können, wurden Lehrende befragt, was für sie eine kreative Leistung von Studierenden ist (BMBF-Projekt „DaVinci“, Teilprojekt hochschuldidaktische Aspekte). Aus den 20 ExpertInneninterviews¹ lassen sich sechs Stufen der Kreativitätsförderung ableiten, die in Tab. 1 dargestellt werden [JH10]. Zur Veranschaulichung der Stufen ist in der rechten Spalte eine Auswahl an Antworten aufgeführt, die Lehrende auf die Frage nach den kreativen Leistungen ihrer Studierenden gegeben haben.

Die sechs Stufen bilden ein breites Spektrum möglicher Ziele zur Kreativitätsförderung ab, was auf heterogene Kreativitätsverständnisse der Lehrenden zurückzuführen ist, die ihrerseits geprägt sind durch unterschiedliche disziplinäre Einflüsse, weltanschauliche Positionen und verschiedenartige Lehr-/Lernsituationen. Für Lehrende in Grundlagenvorlesungen der Physik oder der Betriebswirtschaftslehre ist es beispielsweise ein respektable Erfolg, wenn sie über eine entsprechende Gestaltung der Veranstaltung dazu beitragen, das reflektierende Denken oder selbständige Lernen ihrer Studierenden zu fördern. Lehrende in kleinen Seminaren mit großen Freiräumen in vielleicht sogar gestaltenden Disziplinen wie der Architektur oder Informatik haben hingegen eher die Möglichkeit, in ihren Veranstaltungen eine neue Denkkultur oder die Empfänglichkeit der Studierenden für originelle, gänzlich neue Ideen zu fördern. Dennoch ist auch Letzteres schlecht ohne reflektierend denkende und selbständig arbeitende Lernende möglich, insofern schließen sich die sechs Stufen nicht untereinander aus, sondern bauen vielmehr aufeinander auf.

1 Von 20 sind 10 Lehrende aus Universitäten der Universitätsallianz Metropole Ruhr (UAMR) und 10 Lehrende deutschlandweit (Lehrpreisträger, bei „mein-prof.de“ hoch gerankte Lehrende, und DozentInnen, die das Thema „Kreativität“ explizit zum Gegenstand ihrer Lehre gemacht haben).

Tab. 1: Die sechs Ziele der Kreativitätsförderung

Ziele der Kreativitätsförderung	Beschreibung	Beispiele aus den Interviews „Was ist für Sie eine kreative Leistung Ihrer Studierenden?“
6. Originelle, völlig neue Ideen	Kann nicht erzwungen werden die Möglichkeit des Anflugs vorbereiten Fehler zulassen	andere Lösungswege nutzen/darlegen Stoff für eine Geschichte ausdenken ungewöhnliche, originelle Themen für Hausarbeiten etc.
5. Die Förderung einer neuen Denkkultur	neue Haltung zur Vielperspektivität Reflexion über eigene Kreativität und eigene Denkstruktur (konvergente, divergente Pfade)	Studierende betrachten ein Thema aus mehreren Perspektiven Norm-/Konsensabweichung; sinnvolle Abänderung von Routinen/Regeln Studierende stellen Bezüge zu anderen Disziplinen her
4. Die Förderung kreierenden Lernens	etwas „Schaffen“ Ideen, Thesen, Vernetzungen, Texte, Präsentationen, Forschungsarbeiten, Szenarien u.v.m.	Studierende „schaffen“ etwas zum Thema (Tagungsplanung/-durchführung; e-Infrastruktur-Konzept; Podcast-Beiträge; statt Referat: Gestaltung einer Unterrichtsstunde für LehrerInnen) bisher unbekannte Verknüpfungen werden kreiert
3. Neugier und Begeisterung fördern - Lernmotivation steigern	Abwechslungsreiche Lehre interessante Frage-/ Problemstellungen Reflexion über individuelle Lernmotivation	Praxis-/Erfahrungsbezug ermöglichen Studierende darin fördern, für sich die effektivste Lernmethode herausfinden zu können es gelingt, Studierende zu begeistern
2. Förderung selbständigen Lernens	Lernprozesse eigenverantwortlich steuern eigene Entscheidungen treffen	Thema selbständig suchen Eigene Fragestellungen entwickeln Lücken im Wissensstand aufdecken Studierende recherchieren selbst zum Thema Studierende organisieren ihren Lernprozess selbst Eigene Lernziele formulieren
1. Förderung reflektierenden Denkens	Selbst Wissen erarbeiten inneren Dialog führen Querdenken, Bekanntes hinterfragen	nicht repetitiv kritisches Hinterfragen Vorurteile, Annahmen erkennen wenn jemand mitdenkt Diskussionsbeiträge über Aufgabenstellung hinaus arbeiten

Das Modell lässt sich verwenden, indem sich Lehrende folgende zwei Fragen stellen:

- 1.) Auf welcher Stufe kann und möchte ich die Kreativität meiner Studierenden fördern?
- 2.) Wie kann ich das in meinem Kontext erreichen?

Bei der Beantwortung der zweiten Frage sind mehrere Faktoren zu berücksichtigen, so beispielsweise die Frage, ob die gesamte Veranstaltung über das ganze Semester, nur ein einzelner Veranstaltungstermin oder kleinere Teile einzelner Termine verändert werden sollen, ob die Prüfungen umorganisiert werden können und sollen und schließlich, ob der Einsatz neuer Medien dabei helfen kann, das gewünschte Ziel der Kreativitätsförderung zu erreichen.

Die Erkenntnisse basieren auf 20 geführten Interviews der 1. Erhebungswelle (Lehrende, die explizit mit Kreativität in Zusammenhang zu bringen sind) und der 2. (Lehrende an der UAMR). Zudem wurde das Modell mit einer Online-Befragung aller Lehrenden an

der UAMR (3. Welle) überprüft. Zu Beginn des Fragebogens wurde den Lehrenden die offene Frage gestellt, was für sie eine kreative Leistung ihrer Studierenden ist. Anschließend konnten die Lehrenden ihre (bis zu 3) gegebenen offenen Antworten den 6 Förderzielen (s. Tab. 1) zuordnen (Mehrfachantworten möglich) oder die Option „etwas anderes/trifft nicht zu“ wählen. Die Ergebnisse zeigen, dass die sechs identifizierten Förderziele die heterogenen Kreativitätsverständnisse von Lehrenden abbilden. Nur 1,5% (von n=563) der gegebenen Antworten zur Frage, was eine kreative Leistung von Studierenden ist, wurde als „etwas anderes/trifft nicht zu“ kategorisiert, während die anderen Antworten mindestens einem Förderziel zugeordnet wurden. Außerdem lässt sich bereits erkennen, dass keines der Ziele überflüssig ist – alle wurden von den teilnehmenden Lehrenden gleichermaßen berücksichtigt, ohne dass sich gravierende Unterschiede beobachten ließen.

3 Erfahrung zu einem kreativitätsförderlichen Lehrbeispiel

Die Lehrveranstaltung Informatik und Gesellschaft (IuG) ist seit vielen Jahren eine Pflichtveranstaltung des Diplomstudiengangs Informatik der TU-Dortmund im Umfang von drei SWS, die auch im Rahmen des Bachelorstudiums und des „Studium Fundamentale“ besucht werden kann. Sie wurde bisher als Vorlesung mit begleitenden Übungen durchgeführt. Die Veranstaltung beschäftigt sich mit der Gestaltung sozio-technischer Systeme sowie mit Methoden zur Technikeinführung und Veränderung bestehender IT-Systeme. Ein zentrales Lernziel ist, wie der künftige Nutzungsprozess mit Beteiligung der künftigen Nutzer entwickelt werden kann, um die Arbeitsaufgaben/-Abfolgen mit neuen oder veränderten IT-Lösungen möglichst gut aufeinander abzustimmen. Zudem werden Methoden für die Evaluation von IT-gestützten Arbeitsprozessen erläutert.

Förderung kreierenden Lernens (Stufe 4)

Für die Rekonzipierung der Veranstaltung wurde Stufe 4 des Kreativitätsmodells – das kreierende Lernen – fokussiert. Zentrale Aufgabe war, dass Studierende ein Konzept einer e-Infrastruktur für eine fiktive, private, staatlich-anerkannte Universität in NRW mit dem Namen *NeuDoBoDu*.² entwickelten. Das Projekt – die Erstellung eines Konzeptes für die e-Infrastruktur – wurde in Teilprojekte, wie z.B. Entwurf der universitären Webseiten, Organisation der Lehre und Neuentwicklung einer modernen Bibliothek, aufgeteilt, die jeweils von drei Studierenden bearbeitet wurden. Die Schnittstellen zwischen den Teilprojekten und die Ausarbeitung des gemeinsamen Konzepts wurden vom Teilprojekt „Meta-Koordination“ durchgeführt. Das Besondere an diesem Konzept ist, dass die Informatikstudierenden zunächst selbst erste Ideen entwickeln konnten, dafür auch bestehende Systeme und Literatur recherchieren sollten, dann potentielle NutzerInnen in die Konzipierung einbeziehen mussten (z.B. durch Interviews) und schließlich das finale Konzept erstellten.

Ablauf (Modus der Veranstaltung)

Die Veranstaltung umfasste sechs vierstündige Vorlesungen (jeweils im April, Mai, 2x Juni, 2x Juli; 14:00–18:00 Uhr), fünf einstündige Übungen sowie Coaching-Stunden. Sie wurde von einer Dozentin, einem wissenschaftlichen Mitarbeiter und einer studentischen Hilfskraft betreut. 20 Studierende haben regelmäßig aktiv teilgenommen.

2 Der Name ist in Anlehnung an bestehende Universitäten entstanden: Neu Dortmund Bochum Duisburg.

Jede *Vorlesung* bestand aus einer *Präsentations-Phase*, in der die Studierenden ihre bisher erarbeiteten Ergebnisse vorstellten, einer *Input-Phase*, in der die Dozentin in knapper Form die Inhalte darstellte, die zur Bearbeitung des nächsten Aufgabenschrittes notwendig waren, und einer *Kreativ-Phase*, in der die Studierenden mit Hilfe unterschiedlichen Kreativitäts- und Denktechniken, moderiert durch die Veranstalter, Ideen zur weiteren Arbeit entwickelten. In den *Übungen* stellten die Teilprojekte ihr Vorgehen und ihre bisherigen Ergebnisse als Vorbereitung auf die Präsentation in der nächsten Vorlesung vor. In den *Coaching-Stunden* bekamen die Gruppen Gelegenheit, Fragen persönlich zu klären und ihr Vorgehen zu diskutieren, um den Fokus der Aufgabenstellung zu reflektieren. Unterstützt wurde die Veranstaltung durch das Lernmanagement-System EWS der TU Dortmund, dessen Wiki-Funktion von den Studierenden dazu genutzt werden sollte, Projektaufgaben und -fortschritte zu kommunizieren.

Kreative Ergebnisse aus den Teilprojekten?

Es waren teils sehr kreative Ideen dabei. So hat bspw. das Bibliotheksteam ein System konzipiert, welches die moderne Bibliothek als „warmen Lernraum“ entwickelt. In ihrem Konzept gibt es designte Möbel und Kaffeemöglichkeiten, überall ist WLAN verfügbar, an bestimmten Orten gibt es Teamorte für gemeinsames Lernen und weiterhin ruhige Lese- und PC-Orte. Die Bücher werden entweder in elektronischer Form zur Verfügung gestellt oder per Druckknopf an den entsprechenden Standort geliefert. Die Bücher sind unterirdisch gelagert und ein Fabriklaufband organisiert die Zuordnung zum gewünschten Standort. Ein weiteres kreatives Produkt war das „interaktive Türschild“ über das Lehrveranstaltungen und Räume organisiert werden können, welches auch mit der zentralen Stelle der Lehrveranstaltungsorganisation verbunden ist. Gruppenübergreifend wurde ein individualisierbares Benachrichtigungssystem erdacht, bei welchem eine Benutzerin oder ein Benutzer sich aussuchen kann, wie sie oder er über Neuigkeiten informiert werden möchte (z.B. SMS-Kurznachrichten, E-Mails, Twitter, Facebook oder andere Web 2.0 Dienste). Das „interaktive Türschild“ kann beispielsweise jeden Studierenden über spontane Raumwechsel benachrichtigen.

Lessons learned – qualitatives Feedback

Zum Ende jeder Sitzung wurden moderierte Feedback-Runden durchgeführt. Zudem wurde zum Ende der Veranstaltung im Juli 2010 ein Evaluationsbogen verteilt. Aus den qualitativen Feedbacks konnte viel für die Lehrveranstaltung gelernt werden. Die Teilnehmer fühlten sich bspw. zu Beginn durch die Aufgabenstellung überfordert und ihre Rollen waren ihnen teilweise unklar (Feedback 1. Session). Sie zeigten Unsicherheit im Umgang mit den Moderationsmethoden, so ließen sich z.B. nicht alle Teilnehmer auf die Methode des Brainwriting (2. Session) ein. Trotz ausführlicher Erläuterung zu den empirischen Methoden mit Fallbeispielen, wurde in der 2. Übungsgruppe erneut über einige qualitativ-explorative Methoden (z.B. Interview-Durchführung) diskutiert, was dazu führte, dass geeignete Arbeitsschritte blockiert wurden.

Lessons learned – Stufe 4 erreicht?

Mit einem zusätzlichen quantitativen Beurteilungsinstrument war das Ziel verbunden, festzustellen, ob das Kreativitätsziel der Stufe 4 „Kreierendes Lernen“ (s. Tab. 1) erreicht werden konnte. Dazu wurden die sechs Stufen des Rahmenkonzepts mit jeweils mehreren Items auf einer Likert-Skala mit fünf Antwortmöglichkeiten von „stimme voll zu“ (1) bis „stimme gar nicht zu“ (5) operationalisiert. Von 20 Studierenden haben 19 den Fragebogen ausgefüllt. Im Ergebnis kann festgehalten werden, dass das Ziel der

Kreativitätsförderung auf Stufe 4 aus Sicht der Studierenden erreicht wurde. Die 5 entsprechenden Items wurden im Durchschnitt mit 2,2 bewertet. Aber auch in anderen Stufen wurden Zustimmungen ausgemacht, so zum Teil in Stufe 1 (die überwiegende Zahl der Items wurden positiv bewertet), in Stufe 2 und zum Teil in Stufe 6, während die Förderung von Neugier und Begeisterung (Stufe 3) oder einer neuen Denkkultur (Stufe 5) verfehlt wurde. Da die Stufen 5 und 6 gar nicht als Förderungsziel angestrebt wurde, Stufe 4 aber aus didaktischer Sicht auf den Stufen 1–3 aufbaut, ergibt sich für die dritte Stufe noch ein Verbesserungsbedarf am Lernkonzept.

4 Fazit: kreatives Denken und kreatives Handeln – Vielfalt fördern

Der Beitrag illustriert, wie ein hochschuldidaktisches Konzept (hier: Modell zur Kreativitätsförderung) anhand einer konkreten Lehrveranstaltung (hier Informatik & Gesellschaft) fachdidaktisch erprobt und evaluiert werden kann. Das sechsstufige Modell der Kreativitätsförderung zeigt zudem, wie vielfältig sich die Kreativität von Studierenden fördern lässt. Kreatives Denken und kreatives Handeln bedeutet unter anderem in Handlungssituationen auch neue Möglichkeiten zu bedenken und dementsprechend kompetent, selbstbestimmt und verantwortungsbewusst handeln zu können, und nicht bloß etwas zu rezipieren [ST09]. Mit Bezug zum 6-Stufen-Modell haben wir dargestellt, wie die Vielfalt von kreativem Handeln ausdifferenziert und in Hochschulen – anhand eines konkreten Veranstaltungsbeispiels – gefördert werden kann.

Literatur

- [AK01] Anderson, L. W. & Krathwohl, D. R., et al. (Eds.): A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives. Allyn & Bacon. Boston, MA (Pearson Education Group), 2001.
- [Bl56] Bloom, B.S. & Krathwohl, D. R.: Taxonomy of Educational Objectives: The Classification of Educational Goals, by a committee of college and university examiners. Handbook I: Cognitive Domain. NY, NY: Longmans, Green, 1956.
- [Br06] Brodbeck, K.: Neue Trends in der Kreativitätsforschung. New trends in creativity research, in: Psychologie in Österreich, H. 4&5 (26), S. 246–253, 2006.
- [JH10] Jahnke, I. & Haertel, T.: Kreativitätsförderung in Hochschulen – ein Rahmenkonzept In: Das Hochschulwesen. Jg. 58, H. 3, S. 88–96, 2010
- [ST09] Schulz-Zander, R. & Tulodziecki, G.: Pädagogische Grundlagen für das Online-Lernen. In: L. Issing und P. Klimsa (Hrsg.): Online-Lernen. München: Oldenbourg. 2009. S. 35–45.
- [So07] Sonnenburg, St.: Kooperative Kreativität: theoretische Basisentwürfe und organisationale Erfolgsfaktoren. Wiesbaden: Dt. Univ.-Verl., 2007.
- [St99] Sternberg, R. J.: Handbook of Creativity, New York: Cambridge University Press, 1999.
- [Wa10] Wagner, W.: Tatort Universität. Vom Versagen deutscher Hochschulen und ihrer Rettung: Klett-Cotta, 2010

Fallstudien in medialen Räumen

Jörg Hafer, Joachim Ludwig, Marlen Schumann

Erwachsenenbildung/Weiterbildung und Medienpädagogik – Universität Potsdam
14469 Potsdam – Germany
Web: www.uni-potsdam.de/agelearning
E-Mail: agelearning@uni-potsdam.de

Zusammenfassung: Ziel dieses Beitrages ist es, das didaktische Konzept Fallstudien und seine lerntheoretisch-didaktische Begründung vorzustellen. Es wird die These begründet, dass mediale Räume für die Bearbeitung von Fallstudien lernunterstützend wirken und sich in besonderer Weise für Prozesse der Lernberatung und Lernbegleitung in der Hochschule eignen. Diese These wird entlang dem lerntheoretischen Konzept der Bedeutungsräume von Studierenden in Verbindung mit den Spezifika medialer Räume entfaltet. Für den daraus entstandenen E-Learning-Ansatz Online-Fallstudien kann hier lediglich ein Ausblick gegeben werden.

1 Ausgangslage & Zielsetzung des Beitrages

Handlungsorientierung, Projektorientierung, Kooperation, Praxisbezug und Transferstärke sind relevante didaktische Leitideen. Die Bildungsarbeit mit Fallbeispielen, mehr aber noch mit Fällen bis hin zu Fallstudien berücksichtigen diese Leitideen in besonderem Maße, weil konkrete Handlungssituationen mit theoretischen Bezügen verknüpft werden, was sowohl eine theoretische Herausforderung darstellt als auch praktischen Ertrag abwirft. Durch die Arbeit an Fällen kann Wissen angeeignet, der Umgang mit Wissen geübt und neue Erkenntnisse über den Fall, d.h. zugleich neue Handlungsoptionen, gewonnen werden. Insbesondere im Studium gewinnt die Arbeit mit Fällen an Bedeutung, denn sie fördern nicht nur die Motivation zum Lernen, sondern auch den Praxisbezug der Bildungsmaßnahme. In diesem Beitrag wird das didaktische Konzept Fallstudien skizziert und ein Ausblick zur Umsetzung in medialen Räumen gegeben.

2 Lernen als Erweiterung & Differenzierung von Bedeutungsräumen

Wir knüpfen mit dem Fallstudien-Konzept an eine weithin geteilte Vorstellung von Lernen an, die den Ausgangspunkt von Lernprozessen in schwierigen, herausfordernden Situationen sieht (Ausubel, Bruner, Dewey, Lave und Wenger, Holzkamp, um nur einige zu nennen). Anknüpfend an Klaus Holzkamp (1993) verstehen wir individuelle Lernhandlungen in der Weise, dass sie ihren Ausgangspunkt dann in Handlungsproblematiken nehmen, wenn der Einzelne mit dieser Handlungsproblematik ein Lerninteresse hin auf Erweiterung seiner Handlungsfähigkeit verknüpfen kann. Dabei umfasst Handlungsfähigkeit nicht nur praktische Fertigkeiten, sondern insbesondere die Fähigkeit Handlungszusammenhänge zu verstehen und sich in der Situation verständigen zu können. Der Lernende knüpft im Kontext der Handlungsproblematik, die er zu seiner Lernpro-

blematik macht, an seine vorhandenen und als unzureichend empfundenen Bedeutungs- und Wissenshorizonte an und versucht diese so zu erweitern und zu differenzieren, dass die empfundene Diskrepanzerfahrung aufgehoben und Handlungsfähigkeit (wieder) hergestellt wird. Lernen ist so gesehen im Kern ein Prozess der Bedeutungserweiterung und -differenzierung, der auf die Selbstverständigung/Handlungsfähigkeit in Handlungssituationen zielt. Es stellt sich damit die interessante Frage, wann Studierende eine Handlungsproblematik als Lernherausforderung erleben. Grundsätzlich gilt einerseits zwar, dass die Entwicklung von Lerninteressen und damit die Entwicklung von Lernherausforderungen nicht steuerbar sind. Andererseits sind didaktische Lernarrangements mehr oder weniger geeignet, Lerninteressen entstehen zu lassen. Die Beschreibung praktischer Aufgabenstellungen, Projektarbeit u.ä. bieten den Studierenden Möglichkeiten bei den eigenen Bedeutungs- und Wissenshorizonten anzukoppeln. Lernen vollzieht sich für den Lernenden in sinnhaften Bedeutungsräumen, die an vergangene, gegenwärtige oder antizipierte Situationen gekoppelt sind [Ludwig, Petersheim 2004]. Lernen wird hier als aktiver Prozess der Differenzbildung der Lernenden/Studierenden zwischen individuell verfügbarem und noch fremdem Wissen/Bedeutungshorizonten verstanden. In didaktischer Hinsicht kann dieser Prozess in Form einer Lernberatung und Lernbegleitung durch den Lehrenden unterstützt werden.

Neben dem didaktischen Konzept wird aber auch eine Lernumgebung erforderlich, die eine gewisse Eigentätigkeit zur Entfaltung von Sinnhorizonten, aber auch Chancen für einen Perspektivenaustausch bietet. Aus der Sicht der Informations- und Kommunikationstechnologie ist (fast) alles möglich, aber nur wenig ist wirklich sinnvoll. Denn allein mit der Bereitstellung multimedialer Funktionen ist noch kein Lernfortschritt verbunden [vgl. Ludwig, Petersheim 2004, 265].

„Ein Medium kann Wissen nicht auf den Lernenden "transportieren", sondern Lernprozesse lediglich anregen [vgl. Kerres 2005]. Somit ist der Betrachtungsrahmen zu erweitern, um eine Dimension, die den "E-Lernenden" in den Fokus der Betrachtung rückt, die "Technik" als Mittel zum Zweck und als Form der Lernerunterstützung begriff.“ [Zinth, Schütz In: Holten, Nittel 2010, 98]

2.1 Kooperatives Lernen in medialen Bildungsräumen

In „kooperativen Lernverhältnissen“ [Holzkamp 1993, 510] beziehen Lernende ihre je unterschiedlichen Lernproblematiken entlang eines gemeinsamen Lerngegenstandes aufeinander, wie er durch die Fallzerlegung eingeführt wird. Im Konzept Fallstudien beziehen und beraten Studierende sich, mit Unterstützung eines Fallberaters/Lehrenden, gegenseitig bei ihren Versuchen, ihre individuellen Lerninteressen zu entwickeln.

„Im Kern geht es um die Anerkennung der unterschiedlichen und fremden Bedeutungshorizonte als prinzipiell mögliche, begründete und damit sinnvolle Bedeutungshorizonte. Differenzbildungs- und Selbstverständigungsprozesse werden dann behindert oder gar verhindert, wo sich Objektivismen einschleichen, Bedeutungshorizonte gegen andere durchgesetzt werden sollen und auf diese Weise der potentielle Raum zum leeren Raum wird.“ [vgl. Ludwig, Petersheim 2004]

Dort, wo Anerkennungsverhältnisse sich etablieren und Objektivismen vermieden werden, kann der mediale Raum günstige Voraussetzungen für das Lernen schaffen. Die besondere Lernproblematik des Lernenden wird im virtuellen Raum leichter gemeinsam bearbeitbar und kann durch die anderen Teilnehmenden und die Lehrenden verstanden und beraten werden.

„Die Vorteile des virtuellen Bildungsraumes als kooperativem Lernzusammenhang liegen nach den bisherigen Erfahrungen in der distanzierten Rekonstruktion subjektiver Handlungsproblematiken und Lernbegründungen.“ [Ludwig 2003]

Im Folgenden wird mit den Fallstudien ein didaktisches Konzept der kooperativen Lernbegleitung vorgestellt und ein Ausblick zur Umsetzung im medialen Raum begründet.

Unsere Annahme ist, dass mediale Räume dann zu Bildungsräumen werden können, wenn sie Merkmale des „potentiellen Raumes“ aufweisen, d.h. Räume darstellen, in denen der Lernende mit den Wissenskonstrukten und Vorstellungen experimentell umgehen und Möglichkeiten ausprobieren kann. Soll Lernhandeln unterstützt werden, muss der mediale Raum gegenüber instrumenteller Vereinnahmung „Freiräume“ aufweisen. In der Regel bringen mediale Arrangements, im Sinne von „gestalteten Räumen“, diese Eigenschaften aber nicht einfach mit. Die Möglichkeiten, die mediale Räume zunächst anbieten (z.B. in der Form von Kommunikationswerkzeugen, Internet-Plattformen, Inhalts-Editoren oder Interaktionsschablonen), werden von den Gestalter/-innen (Lehrenden, Weiterbildungsdurchführenden) gestaltet, strukturiert und arrangiert. Erst in dieser „Deutung“ durch die Gestalter/-innen erhalten die Medienelemente die mögliche Funktion eines potentiellen Raums, in dem Lehr-Lern-Prozesse unterstützt werden können.

3 Arbeitsmodell „Fallstudie“

„Fallstudien“ stellen ein Arbeitsmodell dar, um Studierende bei der Entwicklung von Lerninteressen zu unterstützen. Die Generierung von Lerninteressen, was im Hochschulalltag selten Berücksichtigung findet, wird als Handlungsziel des Dozierenden in den Mittelpunkt gestellt. Die Aufgabe des Lehrenden ist es hierbei, aus dem Kontext einer disziplinären Problemstellung eine individuelle Handlungs- und Lernproblematik darzulegen, die dem Lehrenden tatsächlich „passiert“ ist. Ausgangspunkt der Fallstudie ist also eine vom Lehrenden erlebte Handlungs- und Lernproblematik, die er den Studierenden anbietet, damit diese sich damit auseinandersetzen und ihrerseits eine Handlungsproblematik, Anfragen und Lernproblematiken entwickeln können. Der Lehrende führt sich also in einer Art Selbstverständigung seine „früheren“ Lernproblematiken, Zugänge zur disziplinären Problemstellung vor Augen und bereitet sie als Fallerzählung vor. Ziel ist es, den Lernenden das Feld professionellen Denken und Handelns zu öffnen.

Wann passiert was? Ablauf der „Fallstudie“

1. Fallerzählung

Den Ausgangspunkt einer Fallstudie in einer Lehrveranstaltung bildet eine vom Lehrenden eingebrachte und selbsterfahrene Fallerzählung. Ein Fall ist eine komplexe, problematisch empfundene Situation, für die der Lehrende in seiner Biographie bereits eine Lösung gefunden hat. Er stellt seine Problemsicht auf diesen Fall und zugleich seinen Zugang zur Lösung dieses Problems vor. Die Aufgabe des Lehrenden ist es somit, eine disziplinäre Problemstellung als eine individuelle Handlungsproblematik, die zugleich eine relevante Fragestellung der Disziplin ist, in die Lehrveranstaltung einzubringen. Der Lehrende begibt sich mit dem Einbringen einer Fallerzählung, dem didaktischen Angebot einer Fallstudie, in eine Doppelrolle. Er ist einerseits Fallgeber und -erzähler, andererseits aber auch Fallberater und -moderator. Er hat also die nachfolgenden Schritte zu moderieren.

Ein Beispiel aus dem Informatikstudium: Im Rahmen einer Lehrveranstaltung zum Software-Design erzählt die Lehrende von einer Handlungssituation, in der über das Entwicklungsmodell für eine Testsoftware im Automobilbau entschieden werden musste. Das wissenschaftliche Beraterteam, dem sie angehörte, musste in einer komplexen Handlungssituation eine Reihe von Entscheidungen treffen und vertreten. Im Weiteren erzählt sie von einer spezifischen problematischen Situation, in der eine disziplinär relevante Fragestellung (z.B. nach der angemessenen Modellierungssprache für dieses Problem) von Bedeutung war. Die Fall Erzählung (textlich, audio-visuell, grafisch dargestellt), einschließlich ergänzender Ressourcen, wird online verfügbar gemacht. Dafür werden Medienfunktionen eingesetzt, die Kommunikation, Kooperation, Kollaboration und Koordination im Online-Raum ermöglichen [vgl. Keil 2010, 136].

Zusammengefasst geht es um die Erzählung und Dokumentation eines komplexen, realen Falls, in dem die handelnden Personen sichtbar sind und den die Studierenden im Folgenden durch Nachfragen und in Differenzierungsprozessen durchdringen können. Wesentliche Herausforderung für die Umsetzung im medialen Raum ist es, die Entwicklung der Online-Umgebung zu einem gemeinsamen Bedeutungsraum zu unterstützen.

2. Nachfragen

Die Studierenden nähern sich dem Fall durch Nachfragen und eigene Recherchen. Sie sind aufgefordert, Interessierendes oder noch Unklares nachzufragen. Fachwissen wird dabei bereits durch den Lehrenden über seine Erzählungen und seine Sicht/Perspektive auf den Fall eingeführt. Im weiteren Verlauf steht der/die Lehrende als Wissenschaftler/in und Experte/Expertin zur Verfügung, der oder die einen bestimmten Blick auf den Gegenstand anbietet. Die Studierenden haben die Möglichkeit über die Darstellung der individuellen Problematik des Lehrenden hinaus, weitere Recherchen zur disziplinären Problemstellung durchzuführen, um weitere Aspekte – über die individuelle Lernproblematik des Lehrenden hinaus – zu entdecken. Dieser Arbeitsschritt der Fallstudie zielt nicht auf „Problemlösung“ oder „notwendiges Wissen“ im Sinne problemlösungsorientierter Arbeitsmodelle ab, sondern auf vertieftes Verstehen und der Entwicklung eigener „fallangemessener Sinnsysteme“ [Müller 1999] durch die Studierenden.

*Bezogen auf unser Beispiel aus dem Informatikstudium:
Die Studierenden stellen Nachfragen: Wie war der Stand der Entwicklung in den Modellierungssprachen zum Zeitpunkt des Falls? Welche Rolle spielen die Testsysteme für den Automobilbau? Welche theoretischen Grundannahmen liegen den Testverfahren zu Grunde? Wie war das Verhältnis von Wissenschaft und Industrie? Wie würde man das Problem heute lösen? Nachfragen können auch Sachverhalte betreffen, die nicht direkt mit dem Thema der Lehrveranstaltung zusammenhängen. Die Lehrende gibt zu einzelnen Fragen auf das Verstehen des Falls bezogene nähere Erläuterungen bzw. stellt ergänzende Ressourcen zur Verfügung. Nachfragen und Antworten werden im Online-Raum öffentlich und für alle bearbeitbar zugänglich gemacht.*

Die Herausforderung für die Online-Unterstützung ist hier jene, dass jede in begründeter Form formulierte Nachfrage „abgebildet“ wird. Der Sammlung der Nachfrage- und Rechercheperspektiven der Studierenden muss genügend Raum geboten werden, sodass die Studierenden die Möglichkeit besitzen, sich mit ihren persönlichen Bedeutungsperspektiven an den Fall anzukoppeln. Weiterhin müssen in kooperativen Settings die Lernenden jederzeit die Möglichkeit haben, das Lernsetting zu verlassen und autonom zu lernen [vgl. Holzkamp 1993, 513f]. Wichtig wird hier, dass jede/r Studierende seinen persönlichen Raum hat, um arbeiten zu können, Notizen zu machen, Dokumente anzupassen etc. Hier ist es notwendig, Möglichkeiten der Strukturierung und Verknüpfung, sowohl auf der Ebene der Arbeit im Online-Raum der Gruppe als auch für jeden einzelnen Studierenden, zur Verfügung zu stellen.

3. Spuren suchen

Beim „Spuren suchen“ wird der Fall entlang von fünf Dimensionen bearbeitet:

1. Handlungsebene – Das Verstehen der Fallpersonen in ihrem Handeln (Welche guten Gründe und Erkenntnisinteressen hatte der Fallzähler bzw. die beteiligten Personen? Welche Interessen verfolgten die Akteure mit ihrem Handeln?)
2. Beziehungsebene – Das Verstehen der Interaktionsvorgänge und Beziehungen (In welchen Beziehungszusammenhängen standen die Fallpersonen? In welchen Beziehungen stehe ich hier im Studium? Gibt es Parallelen bzw. Differenzen?)
3. Die Ebene der Handlungslogik – Welchem theoretischen Modell folgt die Bearbeitung der Fragestellung durch den Fallzähler? Welche anderen Modelle sind noch möglich?
4. Die Ebene der Rahmenbedingungen – Das Verstehen des Umfeldes (Welche wissenschaftlich-organisatorischen Rahmenbedingungen waren gegeben?)
5. Die Ebene der gesellschaftlichen Rahmenbedingungen – Das Verstehen des gesellschaftlichen Umfeldes (Welche gesellschaftlichen Rahmenbedingungen spiegeln sich in der Fallgeschichte wider?)

Die „Spurensuche“ zielt darauf ab, über eine tiefere Verständigung (über Aspekte) des Falles der Verknüpfung mit je individuellen Bedeutungssystemen und damit auch dem Selbstverständigungsprozess der Studierenden im Hinblick auf ihre möglichen Lerninteressen Raum zu geben.

Bezogen auf unser Beispiel aus dem Informatikstudium:

Zu einigen Themen hat die Dozentin weitere Erläuterungen und ergänzendes Material eingestellt (Testsysteme im Automobilbau, Software-Design und industrielle Anforderungen). Weitere Themen wurden von Studierenden als interessant identifiziert. Der Online-Raum wird genutzt, um die vorhandenen Ressourcen und Dokumente zu strukturieren und ggf. gemeinsam zu bearbeiten. An diesem Punkt ist eine Online-Moderation verfügbar. Dafür werden diverse Medienwerkzeuge, die für Kommunikation (Forum, Chat, Virtuelle Arbeitsräume) und Kooperation/Kollaboration (Wiki, kollaborativer Texteditor; Social Tagging & Bookmarking etc.) geeignet sind, genutzt.

4. Kernthemen sammeln

Aus der Summe von Dargestelltem und Recherchiertem sammeln die Lernenden Kernthemen, welche unkommentiert geclustert und anschließend bearbeitet werden. Kernthemen sind solche Aspekte der Fallstudie, die Studierende in besonderer Weise am Fall interessiert haben. Kernthemen stellen die Lerninteressen dar, die Studierende mit Blick auf die Fallzählung generiert haben.

Bezogen auf unser Beispiel aus dem Informatikstudium:

Das Ergebnis der Sammlung von Kernthemen wird dokumentiert und mit den relevanten Ressourcen verknüpft. Diese strukturierte Sammlung ist die inhaltliche Grundlage der weiteren Arbeit.

5. Kernthemen bearbeiten

Mit den Kernthemen werden Themen identifiziert, die, da sie von den Lernenden zusammengetragen wurden, im Interessensraum der Gruppe liegen. Diese Kernthemen können thematisch gebündelt und in Arbeitsgruppen bearbeitet werden. Am Ende werden die verschiedenen Ergebnisse zusammengeführt und auf den Fall rückbezogen. Die Herausforderung für die online-unterstützte Umsetzung in den Arbeitsschritten 3–5 ergibt sich

durch die Verknüpfung des neu eingebrachten Wissens und den angebrachten Perspektiven auf den Fall als Fallstudie. Wesentlich ist hierbei die Beziehung zwischen den Perspektiven der Studierenden und den Kernthemen.

Zusammenfassend kann gesagt werden, dass das didaktische Konzept der Online-Fallstudie Herausforderungen und Anforderungen für die Gestaltung des medialen Raumes stellt und somit Überlegungen für die technische Umsetzung nach sich ziehen muss, vor allem bzgl. der kommunikativen und kollaborativen Einsatzbereiche digitaler Medien. So scheint zum Beispiel, die Verbindlichkeit in einem medialen Raum im Vergleich zur Präsenzsituation geringer zu sein. Jedoch kann gerade auch der Aspekt der Abgeschlossenheit/Trennung von der Alltagspraxis eben jenen Raum schaffen, um eine ausgegliederte Handlungsproblematik bearbeiten zu können. Das Stichwort Verbindlichkeit wird im didaktisch-konzeptuellen Rahmen des Online-Fallstudienkonzeptes ernstgenommen. Ein notwendiger Aspekt hierbei ist das Personalisieren der Ausführungen, um einen vertrauenswürdigen Rahmen für alle Beteiligten schaffen zu können. Wenn darüber hinaus die Interessen möglichst aller Teilnehmenden Berücksichtigung finden, kann man auf eine engagierte Teilnahme hoffen.

Literatur

- [K10] Keil, Reinhard (2010): E-Learning 2.0 vom Kopf auf die Füße gestellt. In: Herzig, Bardo; Meister, Dorothee M.; Moser, Heinz; Niesyto, Horst (Hg.): Jahrbuch Medienpädagogik 8. Medienkompetenz und Web 2.0. 1. Aufl. Wiesbaden: VS Verlag für Sozialwissenschaften / GWV Fachverlage GmbH Wiesbaden, S. 121–146.
- [H93] Holzkamp, Klaus (1993): Lernen. Subjektwissenschaftliche Grundlegung. Frankfurt, New York.
- [K05] Kerres, Michael (2005): Didaktisches Design und eLearning: Zur didaktischen Transformation von Wissen in mediengestützte Lernangebote. In: Miller, D. (Hrsg.): eLearning. Eine Multiperspektivische Standortbestimmung. Bern u.a.
- [L03] Ludwig, Joachim (2003): Lehr-, Lernprozesse im virtuellen Bildungsraum: vermitteln, ermöglichen, verstehen. In: Arnold, Rolf; Schübler, Ingeborg (Hrsg.): Ermöglichungsdidaktik – erwachsenenpädagogische Grundlagen und Erfahrungen. Baltmannsweiler. S. 262–275. Postprints der Universität Potsdam: Humanwissenschaftliche Reihe 15. (WWW <http://www.projekt-be-online.de/veroeffentlichungen/pdf/virtuellerbildungsraum.pdf>, Stand 3.08.2010)
- [L04] Ludwig, Joachim; Petersheim, Albert (2004): Virtuelle Bildungsräume als Brücke zwischen Lernen und Handeln. In: Bender, W./Groß, M./Heglmeier, H. (Hrsg.): Lernen und Handeln – Eine Grundfrage der Erwachsenenbildung. Schwalbach/Ts. 2004, S. 254–270.
- [M99] Müller, Kurt R. (1999): Das Bildungskonzept „Fallarbeit“ entwickeln und gestalten. München: Bayrisches Staatsministerium für Arbeit und Sozialordnung, Familie, Frauen und Gesundheit.
- [ZS10] Zinth, Claas-Philip; Schütz, Julia (2010): E-Learning in der Hochschulpraxis: Wie Lehren und Lernen nicht auf der (virtuellen) Strecke bleiben. In: Holten, Roland; Nitte, Dieter (Hg.): E-Learning in Hochschule und Weiterbildung. Einsatzchancen und Erfahrungen. Bielefeld. S. 95–106.

KoProV – Ein Lehransatz zur koordinierten Projektvorlesung auf Basis von Wissensmodulen

Christoph Laroque¹, Jonas Schulte¹ und Diana Urban²

¹ Heinz Nixdorf Institut der Universität Paderborn

² Stabsstelle Bildungsinnovationen und Hochschuldidaktik
33102 Paderborn – Germany

Web: <http://www.whni.uni-paderborn.de/>

Email: laroque@uni-paderborn.de | schulte@uni-paderborn.de | diana.urban@uni-paderborn.de

Zusammenfassung: In der universitären Lehre ändert sich der Leitgedanke von einer qualifikationsorientierten hin zu einer kompetenzorientierten Ausbildung. Der Begriff Kompetenz lässt sich dabei grob in die fachlichen und die überfachlichen Kompetenzen unterteilen. Insbesondere die Vermittlung von Schlüsselqualifikationen hat in der Lehre von naturwissenschaftlichen Fachrichtungen nur unzureichend Einzug erhalten. Während der klassische Vorlesungsbetrieb auf den Erwerb von Fachkompetenz zielt, stoßen ausschließlich projektorientierte Veranstaltungen schnell an ihre Grenzen hinsichtlich der Teilnehmergröße oder Umfang der Lerninhalte. Um auf geeignete Art und Weise den Erwerb von überfachlichen Kompetenzen zu ermöglichen, bedarf es neuer didaktischer Konzepte, die eine engere Verknüpfung von klassischen Vorlesungen und dem projektorientierten Lernen vorsehen. In diesem Sinne versucht der skizzierte Ansatz der koordinierten Projektvorlesung (KoProV) Wissensvermittlung im Rahmen von Vorlesungseinheiten mit koordinierten Praxisphasen in Teilgruppen zu verbinden. Für eine erfolgreiche Durchführung und Erarbeitung des begleitenden Praxisprojektes durch mehrere Teilgruppen sind organisatorische und technische Randbedingungen zu beachten.

1 Motivation

Das Handlungs- und Forschungsfeld der (Wirtschafts-)Informatik hat sich in den letzten Jahren grundlegend geändert. Wurden früher eher Anwendungen mit einem überschaubaren Umfang und sehr speziellem Einsatzgebiet entwickelt, so ist heutzutage zumeist die Entwicklung hochgradig integrierter Softwarekomponenten für bestehende IT-Infrastrukturen erforderlich. Die Fähigkeit ein System zu entwickeln, welches sich in bestehende Applikationslandschaften leicht integrieren lässt, ist in der zeitgemäßen Softwareentwicklung unerlässlich. Neben der Verschiebung der Anforderungen an die Softwareentwicklung wird aus hochschuldidaktischer Sicht insbesondere eine stärker kompetenzorientierte Ausbildung in der Universitätslehre gefordert. Speziell in naturwissenschaftlichen Studiengängen ist in der Vergangenheit auf die Vermittlung von Schlüsselkompetenzen, z.B. Methoden-, Sozial- und Personalkompetenz, nur selten eingegangen worden. Im klassischen Vorlesungs-/Übungsbetrieb können diese Handlungskompetenzen nur unzureichend vermittelt werden, da zumeist der Übungsbetrieb mit identischen Arbeitsaufgaben für Einzelpersonen bzw. Gruppen gestaltet wird. Eine Koordination und Absprache zwischen den Teilnehmern wird in dieser Lehr- und Lernform nicht voraus-

gesetzt. Ausgehend davon haben sich diverse Methoden des projektorientierten Lernens¹ (PBL) etabliert [S2003]. Um Studierenden darüber hinaus eine effektive Unterstützung zu gewährleisten, werden häufig kooperationsunterstützende Systeme (KuS) herangezogen, um eine Lernumgebung zu schaffen, in der Studierende selbstkoordiniert ihre Projektarbeit in der Gruppe organisieren können [JJ2009], [MS2009].

An diesem Punkt versucht das Konzept der koordinierten Projektvorlesung (KoProV) anzusetzen, in dem eine Vorlesung mit koordinierten Projektphasen konzipiert wird. KoProV repräsentiert eine Veranstaltung im Umfang von fünf ECTS Punkten, was einer klassischen Veranstaltungsform mit zweistündiger Vorlesung und einstündiger Übung entspricht. Mit diesem neuen didaktischen Ansatz sollen die verschiedenen Anforderungen miteinander verknüpft werden und die Vermittlung von Fach- und Handlungskompetenzen verschmolzen werden. Dabei geht es nicht nur um eine Veränderung der Struktur- und Organisationsform einer Lehreinheit/eines Moduls, vielmehr sollen auch Inhalte und Methoden so neu ausgerichtet werden, dass sie einer kompetenzorientierten Lehre gerecht werden. Dieses Lehr-/Lernformat stellt insofern besondere Anforderungen an den Einsatz von KuS und eLearning-Systemen dar, als es hier häufig zum Einsatz von Spezialsoftware kommt. Um eine durchgängige Unterstützung der Projektarbeit zu gewährleisten, kann daher nicht ausschließlich auf eine einzelne Lernplattform zurückgegriffen werden, sondern es müssen unterschiedlichste Systeme betrachtet und integriert werden.

Angewendet wird das hier vorgestellte Konzept erstmalig zum Wintersemester 2010/2011 im Rahmen einer Bachelorveranstaltung „Simulation von Materialflusssystemen“, die sich anwendungsnah mit dem methodischen Ansatz Simulation im Rahmen des Studiums der Wirtschaftsinformatik an der Universität Paderborn auseinandersetzt. Zugelassen sind neben Bachelorstudenten aus dem Bereich der Wirtschaftsinformatik aber auch Studierende der Wirtschaftswissenschaften, des Wirtschaftsingenieurwesens und der Informatik mit Nebenfach Wirtschaft.

Der Beitrag betrachtet im Folgenden die projektorientierten Lern-/Lehrformen und stellt das neue Konzept der KoProV vor. Die technische und didaktische Modularisierung der Lehr-/Lerninhalte (Abschnitt 3) sowie der konkrete Ablauf einer KoProV-Veranstaltung (Abschnitt 4) bilden den Kern des Artikels.

2 Projektorientierte Lehr- und Lernformen

Projektorientierte Lehr- und Lernformen haben in den letzten Jahren insbesondere in der (Wirtschafts-)Informatik an Bedeutung gewonnen, um berufliche Handlungskompetenzen zu fördern und Studierende so optimal mit den eingeforderten Schlüsselqualifikationen auszubilden. Gängige Lehr- und Lernformen zur Unterstützung der projektorientierten Lehre sind z.B. Projektgruppen, Praktika und Abschlussarbeiten. Die genannten Lehr- und Lernformen ermöglichen es Studierenden in besonderer Art und Weise auf reale Projektarbeit vorzubereiten. Insbesondere wird eine Verbindung von Wissenserwerb mit dem Erwerb bzw. dem Erlernen von Lernstrategien ermöglicht [W2000]. Weiterhin werden fachübergreifende Kompetenzen durch selbstgesteuertes und organisiertes Lernen durch projektorientierte Lehr- und Lernformen gefördert [MF2006]. Jedoch gibt es einige Rahmenbedingungen, die beachtet werden müssen, und innerhalb derer sich die Veranstaltungsplanung situieren muss. Ein Ausschlusskriterium ist häufig die Teilnehmer-

¹ Projektorientiertes Lernen, engl. Project-Based Learning (PBL).

zahl, welche bei den oben genannten Veranstaltungsformen zumeist stark begrenzt ist. Speziell in den Veranstaltungen des Bachelorstudiengangs ist jedoch mit einer hohen Anzahl an Studierenden zu rechnen. Notwendigerweise müsste eine Teilnehmerbegrenzung für die jeweilige Veranstaltung eingeführt werden. Diese korreliert jedoch keineswegs mit der Forderung nach schnelleren Abschlüssen, da es dazu kommen kann, dass Studierende ihre Pflichtveranstaltungen nicht belegen können. Als Konsequenz werden die Bachelorveranstaltungen überwiegend als klassische Vorlesung ohne explizite Teilnehmerbegrenzung durchgeführt und erst im Rahmen des Masterstudiums kommen projektbasierte Lehr- und Lernformen verstärkt zum Einsatz. Jedoch ist ein zentrales Element und Qualitätsmerkmal der Bildung für nachhaltige Entwicklung insbesondere die Partizipation der Studierenden. Ohne aktive Teilhabe und Mitgestaltung von Einzelnen und Gruppen ist eine nachhaltige Entwicklung global wie lokal nicht zu realisieren. Partizipatives Lernen wird aber gerade durch projektorientierte Lehr-/Lernformen gefördert und darf daher keineswegs in den Grund- und Bachelorveranstaltungen des Studiums fehlen.

Peter Drucker hat bereits 1969 den Begriff der Wissensarbeit eingeführt, um zu betonen, dass viele Arbeitsprozesse in Unternehmen zunehmend weniger konventionalisierbar und automatisierbar, sondern wissensbasiert sind [D1969]. Studierenden muss also der Rahmen geschaffen werden, um sich Wissen konstruktiv zu erarbeiten. Weiterhin haben Untersuchungen gezeigt, dass Wissen bei den meisten Personen nur dann nachhaltig erschlossen und verstanden wird, wenn ihnen die Möglichkeit zur Reflexion und Anwendung der Lerninhalte gegeben wird [HM1992].

Um einen nachhaltigen Mehrwert für die Studierenden der (Wirtschafts-)Informatik erzielen zu können, müssen neue didaktische Konzepte erarbeitet werden, welche die Vorzüge von Vorlesungen mit denen der projektorientierten Arbeitsweise der (Wirtschafts-)Informatik verbinden. Wissensvermittlung als Vorlesung soll sinnvoll mit der Wissenserschließung und ko-aktiver Wissensarbeit in Projekten verknüpft werden. In diesem Sinne versucht der Ansatz der KoProV diese Lücke zu schließen, indem die Trennung von Vorlesung und Übung aufgehoben wird und eine kombinierte Form entwickelt wird. Bei der Konzeption eines neuen didaktischen Ansatzes ist es zwingend erforderlich, die unterschiedlichen Lerntypen zu beachten und individuell auf diese einzugehen. Eine Typologie bieten Honey und Mumford. Das neue Arrangement der Lernsituation bzw. -umgebung in Form der KoProV erfordert zudem eine flexible Gestaltung der Lehrendenrolle – vom Wissensvermittler bis hin zum Lernberater.

3 Modularisierung der Lerninhalte

Die Entwicklung und Anwendung von Theorien, Konzepten, Modellen, Methoden und Werkzeugen für die Analyse, Gestaltung und Nutzung von Informations- und Kommunikationssystemen steht im Mittelpunkt des Wirtschaftsinformatik-Studiums an der Universität Paderborn. Dabei greift die Wirtschaftsinformatik vor allem auf Ansätze der Betriebswirtschaftslehre und der Informatik zurück, die sie erweitert, integriert und um eigene spezifische Ansätze ergänzt. Im Rahmen der Lehrveranstaltungen wird den Studierenden in Paderborn in Wahlpflichtveranstaltungen auch spezielles Methodenwissen vermittelt. Eine dieser Methodenveranstaltungen, die im Bachelorstudium als Grundlagenveranstaltung und im Masterbereich als Vertiefungsveranstaltung angeboten wird, zielt auf die Vermittlung grundlegender Kenntnisse der Simulation, im speziellen der Ablaufsimulation von Fertigungssystemen. Simulation ist eine in fast allen wissenschaftlichen

Disziplinen eingesetzte Analysemethode zur Unterstützung der Planung oder Optimierung komplexer (Real-) Systeme [DL2008].

In der geplanten koordinierten Projektvorlesung (KoProV) sollen den Studierenden im Rahmen der Bachelorveranstaltung „Einführung in die Simulation von Materialflusssystemen“ Grundlagen der Methode Simulation, ihre Abgrenzung zu anderen Methoden und erste Anwendungsfälle praxisgerecht vermittelt werden. Als weiterer Aufbau soll ein vertiefendes Modul „Techniken der Materialflusssimulation“ im Masterbereich angeboten werden, das spezielle Fachkenntnisse vermittelt und aktuelle Forschungsthemen behandelt. Die Studierenden sollen hier mit ihrer Arbeit einen eigenen Beitrag zur Forschung in dem adressierten Themenfeld leisten und somit neben der reinen Fachkompetenz weitere Fähigkeiten im wissenschaftlichen Arbeiten und der Präsentation eigenen Wissens erwerben. Die Zusammenführung beider Veranstaltungen sowie eine geeignete Modularisierung von Lerninhalten für den didaktischen Ansatz der koordinierten Projektvorlesung kann technisch durch das Konzept der virtuellen Wissensräume unterstützt werden [H2001].

Aus didaktischer Sicht schließen sich zwei Fragestellungen an:

- Wie kann die Zukunftsfähigkeit der entwickelten Materialien im Alltag gesichert werden?
- Wie kann der entwickelte Content didaktisch sinnvoll in den unterschiedlichsten Lernarrangements integriert werden?

Dafür wurden zwischen 2000 und 2004 verschiedene Contentprojekte vom Bund mit einem Gesamtvolumen von 186 Mio.€ gefördert (vgl. [B2004]).

Um sich den beiden Fragestellungen zu nähern, werden zunächst die Konzepte *Learning Object Metadata (LOM)* und *Reusable Learning Objects (RLOs)* betrachtet. Der LOM Standard ist eine Spezifikation der Syntax und der Semantik zur Beschreibung von Lernobjekten [L2002]. Der Standard betrachtet dabei nicht, wie die Funktionalitäten technisch realisiert werden können. Ein Problem in Zusammenhang mit LOM ist die Granularität der Lernobjekte. Peter Baumgartner beschreibt mit dem Konzept der RLOs eine Möglichkeit, das Zusammenwirken von didaktischen Szenarien und Interaktionsmustern zu ermöglichen. Dabei werden auf einer unteren Ebene pädagogische Interaktionsmuster identifiziert, die auf einer höheren Ebene in Lernszenarien einfließen. Die Idee dabei ist, kleine universell einsetzbare Interaktionsmuster zu identifizieren, deren Wiederverwendung in unterschiedlichen Kontexten möglich ist.

Durch eine entsprechende Modularisierung von Lehrinhalten können im Gegensatz zu gewöhnlichen *Lern-Management-Systemen (LMS)*, in denen der Kurs die kleinste verwaltbare Einheit darstellt, Kurse schneller und unkompliziert mit entsprechenden Inhalten und Informationen gefüllt werden. Das in Abbildung 1 gezeigte Autodesk-Content-Modell stellt ein fünfschichtiges Modell zur Strukturierung von Objekten dar. Einzelne Inhaltsobjekte werden auf höheren Schichten flexibel zu wieder verwendbaren Informations- und Lernobjekten aggregiert und können auf diese Art und Weise in unterschiedlichen Lehrveranstaltungen zum Einsatz kommen. In [B2005] wird sich der Fragestellung der Wiederverwendung von Lernobjekten aus didaktischer Sicht genähert.



Abb. 1: Das Autodesk-Content-Modell nach [M2002].

4 Strukturierung und Ablauf einer KoProV

Das didaktische Konzept der koordinierten Projektvorlesung löst die klassische Trennung zwischen Vorlesung und Übung auf und versucht diese Bestandteile einer Veranstaltung ineinander zu verzahnen. Eine Unterscheidung zwischen einer Vorlesungs- und einer Gruppenarbeitseinheit im Sinne der KoProV ist der Umfang an neuen Lerninhalten, den die Studierenden während der Veranstaltung von dem Dozenten bzw. der Dozentin erhalten. Vorlesungseinheiten sind darauf ausgerichtet, Lerninhalte konstruktiv zu erarbeiten. Der Ansatz der KoProV erlaubt es in besonderem Maße, neue Informationen mit vorhandenen Wissensständen in Beziehung zu setzen, da die Studierenden die Möglichkeit haben, aktiv die Vorlesungsinhalte in einem konkreten Projekt anzuwenden. Dies erfolgt in zweierlei Hinsicht – einerseits werden die Informationen aus Vorlesungseinheiten für die Durchführung in koordinierten Gruppenarbeitseinheiten benötigt, andererseits stellen Fragen und Probleme, die im Rahmen der Gruppenarbeitseinheiten entstanden sind, Diskussionsthemen für eine folgende Vorlesungseinheit dar. Abbildung 2 veranschaulicht das Zusammenspiel aus konstruktiven Vorlesungs- und koordinierten Gruppenarbeitseinheiten.

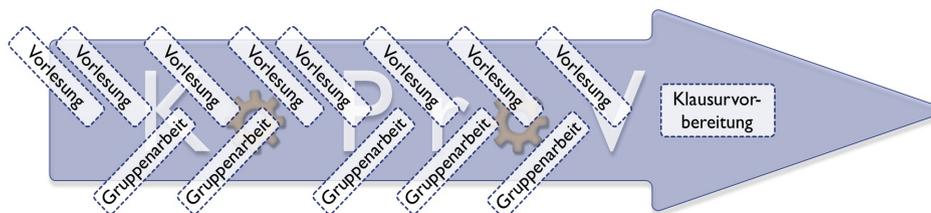


Abb. 2: Ablauf einer koordinierten Projektvorlesung.

Der Ansatz der koordinierten Projektvorlesung grenzt sich insofern gegenüber anderen didaktischen Lehr-/Lernkonzepten ab, als hier insbesondere die kontinuierliche Projektarbeit gefördert wird. Das Konzept des Medi@Thing [K2008] sieht ebenfalls eine gänzliche Auflösung der klassischen Trennung zwischen Übung und Vorlesung vor. Dabei legt das Konzept besonderen Wert auf die gemeinsame Wissensarbeit in der universitären Informatikausbildung. Studierende erarbeiten bei diesem Ansatz ein Thema in Kleingruppen und präsentieren dieses in der großen Runde. Medi@Thing ist damit insbesondere auch für die Ausarbeitungen von Themen geeignet, die nicht im Rahmen der Vorlesung durch den Dozenten/die Dozentin vermittelt werden. Im Gegensatz dazu ist bei der Durchführung von KoProV eine enge Bindung zwischen Vorlesungseinheiten und Gruppenarbeitseinheiten unerlässlich. Zudem berücksichtigt der Ansatz der KoProV Veranstaltung die Koordinierung zwischen einzelnen Teilgruppen. Dies ist in sofern wichtig, als Projekte in der (Wirtschafts-)Informatik zumeist aus mehreren Teilprojekten bestehen, die aufeinander abgestimmt werden müssen. Für die optimale Unterstützung dieser Veranstaltungsform wird empfohlen, kooperationsunterstützende Systeme einzusetzen, damit Projektrichtlinien festgehalten werden können und Informationen zwischen den Teilnehmern und Teilgruppen sowohl orts- als auch zeitunabhängig ausgetauscht werden können. Im folgenden Abschnitt wird auf mögliche technische Unterstützungsformen genauer eingegangen.

4.1 Systemunterstützung für KoProV

Für die Durchführung der Veranstaltung wird eine Systemunterstützung benötigt, welche folgende Aufgaben übernimmt:

1. *Dokumentenmanagement*: Sowohl die Bereitstellung von Vorlesungsmaterialien durch den Dozenten als auch der Austausch von Richtlinien und Schnittstellenbeschreibungen zwischen den Teilgruppen muss gewährleistet sein.
2. *Austauschplattform*: Den Studierenden soll mit Hilfe der Plattform die Möglichkeit gegeben werden, sich in Form von Foren auszutauschen und auch über Gruppen hinweg auszutauschen.
3. *Informationsbibliothek*: Da die Vorlesung von einer heterogenen Gruppe an Studierenden besucht wird, die ihrerseits unterschiedlich stark ausgeprägtes Vorwissen mitbringen, muss das Basiswissen in Form von Skripten etc. bereitgestellt werden. Dabei ist eine spezielle Herausforderung die Integration bestehender Systeme, wie etwa eines Wiki zum Materialflusssimulator *d³Fact insight*.

5 Ausblick

Mit dem neuen didaktischen Ansatz der koordinierten Projektvorlesung ist eine Möglichkeit gegeben, um die Vorzüge von klassischen Vorlesungen mit projektorientierten Lehr- und Lernformen zu verknüpfen. Im Rahmen der Durchführung der ersten KoProV-Veranstaltung, sollen die Auswirkungen auf den Lernerfolg der Studierenden evaluiert werden. Dabei sollen einerseits die Klausurnoten mit denen vorheriger Semester verglichen werden und parallel eine Befragung der Studierenden über das Lehr- und Lernkonzept erfolgen. Weiterhin wird die reale Durchführung der Veranstaltung konkrete Anforderungen an die technische Infrastruktur stellen.

Literatur

- [B2004] Baumgartner, Peter (2004): *Didaktik und Reusable Learning Objects (RLO's)*. In: Campus 2004 - Kommen die digitalen Medien an den Hochschulen in die Jahre? Hg: Doris Carstensen und Beate Barrios. Münster, Waxmann, S. 311–327.
- [B2005] Baumgartner, P. und Kalz, M. (2005): *Wiederverwendung von Lernobjekten aus didaktischer Sicht*. In D. Tavangarian & K. Nölting, ed.: *Auf zu neuen Ufern*. Tagungsband der Jahrestagung der Gesellschaft für Medien in der Wissenschaft (GMW), Waxmann, Münster, Bd. 34, S. 97–106.
- [D1969] Drucker, P.: *The Age of Discontinuity: Guidelines to Our Changing Society*. New York: Harper & Row, 1969. Deutsch: *Die Zukunft bewältigen. Aufgaben und Chancen im Zeitalter der Ungewißheit*, Düsseldorf: Econ, 1969 & 1998.
- [DL2008] Dangelmeier, W. und Laroque, Ch. (2008): *Simulation*. In Kurbel, K; Becker, J; Gronau, N; Sinz, E. J. & Suhl, L.: *Enzyklopädie der Wirtschaftsinformatik*, online verfügbar: <http://www.oldenbourg.de/8080/wi-enzyklopaedie/lexikon/technologien-methoden/Operations-Research/Simulation/index.html/?searchterm=simulation>, Stand: 12.08.2010.
- [H2001] Hampel, T. (2001): *Virtuelle Wissensräume – Ein Ansatz für die kooperative Wissensorganisation*, Universität Paderborn.
- [HM1992] Honey P, Mumford A.: *The Manual of Learning Styles*. 3rd Ed. Maidenhead: Peter Honey, 1992.
- [JJ2009] Jeremić, Z.; Jovanović, J.; Gašević, D. & Hatala, M. (2009): *Project-Based Collaborative Learning Environment with Context-Aware Educational Services*, Learning in the Synergy of Multiple Disciplines, S. 441–446.
- [K2008] Keil, R. (2008): *Medi@Thing – Ein didaktischer Ansatz zum ko-aktiven Lernen*. In: Tagungsband zur 3. GI-Fachtagung Hochschuldidaktik Informatik. Potsdam, Germany. Dezember 2008.
- [L2002] IEEE LTSC (2002): *Learning Object Metadata (LOM) (1484.12.1)*.
- [M2002] Masie, C. (2002): *S3-Guide: Making Sense of Learning Specification & Standards*. A Decision Maker's Guide to their Application.
- [MF2006] Mandl, H und Friedrich, H. F.: *Handbuch Lernstrategien*. Göttingen: Hogrefe, Göttingen 2006.
- [MS2009] Meyer, P.; Sporer, T. & Metscher, J. (2009): *e3-Portfolio - Supporting and Assessing Project-Based Learning in Higher Education via E-Portfolios*. In Ulrike Cress; Vania Dimitrova & Marcus Specht, ed.: *EC-TEL*, Springer, S. 806–810.
- [S2003] Solomon, G.: *Project-Based Learning: a Primer.*, TechLearning. 15.01.2003. Elektronisch verfügbar: <http://www.techlearning.com/article/17708>, Stand: 12.08.2010.
- [W2000] Wild, K.-P. (2000): *Lernstrategien im Studium*. Strukturen und Bedingungen. Münster: Waxmann.

Bisher in dieser Reihe erschienene Bände:

- 1 Andreas Schwill (Hrsg.): Hochschuldidaktik der Informatik. HDI2008 – 3. Workshop des GI-Fachbereichs Ausbildung und Beruf / Didaktik der Informatik 2008
2009 | ISBN 978-3-940793-75-1
- 2 Stechert, Peer: Fachdidaktische Diskussion von Informatiksystemen und der Kompetenzentwicklung im Informatikunterricht
2009 | ISBN 978-3-86956-024-3
- 3 Freischlad, Stefan: Entwicklung und Erprobung des Didaktischen Systems Internetworking im Informatikunterricht
2010 | ISBN 978-3-86956-058-8
- 4 Engbring, D., Keil, R., Magenheimer, J., Selke, H. (Hrsg.): HDI2010 – Tagungsband der 4. Fachtagung zur „Hochschuldidaktik Informatik“
2010 | ISBN 978-3-86956-100-4

In dieser Reihe erscheinen Tagungsbände und ausgewählte Forschungsberichte zu Themen aus der Didaktik der Informatik in Schule und Hochschule.

Mit der 4. Tagung zur Hochschuldidaktik Informatik wird eine Reihe fortgesetzt, die ihren Anfang 1998 in Stuttgart unter der Überschrift „Informatik und Ausbildung“ genommen hat. Seither dienen diese Tagungen den Lehrenden im Bereich der Hochschulinformatik als Forum der Information und des Diskurses über aktuelle didaktische und bildungspolitische Entwicklungen im Bereich der Informatikausbildung. Aktuell zählen dazu insbesondere Fragen der Bildungsrelevanz informatischer Inhalte und der Herausforderung durch eine stärkere Kompetenzorientierung in der Informatik.

Die eingereichten Beiträge zur HDI 2010 in Paderborn veranschaulichen unterschiedliche Bemühungen, sich mit relevanten Problemen der Informatikdidaktik an Hochschulen in Deutschland (und z. T. auch im Ausland) auseinanderzusetzen. Aus der Breite des Spektrums der Einreichungen ergaben sich zugleich Probleme bei der Begutachtung. Letztlich konnten von den zahlreichen Einreichungen nur drei die Gutachter so überzeugen, dass sie uneingeschränkt in ihrer Langfassung akzeptiert wurden. Neun weitere Einreichungen waren trotz Kritik überwiegend positiv begutachtet worden, so dass wir diese als Kurzfassung bzw. Diskussionspapier in die Tagung aufgenommen haben.