Programmieren zur Lösungseingabe in Selbsttests

Dominik Giel

Hochschule Offenburg, Badstraße 24, 77652 Offenburg
dominik.giel@hs-offenburg.de

Kurzfassung

Selbsttests in Lernmanagementsystemen (LMS) ermöglichen es Studierenden, den eigenen Lernfortschritt einzuschätzen. Im Gegensatz zur Einreichung und Korrektur vollständig ausformulierter Aufgabenlösungen nutzen LMS überwiegend die Eingabe der Lösung im Antwort-Auswahl-Verfahren (Single-Choice). Nach didaktischen Ansatz „Physik durch Informatik“ geben die Lernenden stattdessen ihre Aufgabenlösungen in einer Programmiersprache ins LMS ein, was eine automatisierte Rückmeldung erleichtert und das Erreichen einer höheren Kompetenzstufe fördert. Es wurden zehn LMS-Selbsttests erstellt, bei denen die Lösungen zu einer Lehrbuch-Aufgabenstellung jeweils durch Eingabe in einer Programmiersprache und von einer Kontrollgruppe im Antwort-Auswahl-Verfahren abgefragt wurden. Ergebnisse aus dem ersten Einsatz dieser Selbsttests für die Lehrveranstaltung Physik im Studiengang Biotechnologie werden vorgestellt.

1. Zielsetzung

In der Studieneingangsphase ist die Physikvorlesung Teil der meisten Ingenieurstudiengänge. Die angehenden Ingenieure erwerben in der Physikvorlesung die Kompetenz zur Nutzung mathematischer Modelle für die Darstellung technischer Vorgänge. Dies geschieht in der Regel anhand einfacher Beispiele aus Mechanik, Wärmelehre und Elektrizitätslehre. Die Aufgabenstellungen umfassen zum Beispiel überlagerte Bewegungen (Schiefer Wurf), Kräftezerlegungen (Schiefe Ebene) und einfache Drehbewegungen (Starrer Rotator). Diese Themen bereiten vertiefende Vorlesungen wie technische Mechanik, Thermodynamik und Strömungslehre vor, indem sie die Anwendung mathematischer Methoden (Gleichungssysteme, Differenzialgleichungen, Extremwertprobleme und Fehlerfortpflanzung) anhand einfacher Beispiele erläutern. Aus Studierendensicht legt diese Fokussierung auf wenige, einfache Modelle das Auswendiglernen von Lösungsformeln nahe, was aus didaktischer Sicht nicht das erwünschte Lernziel ist. Konventionelle Lernmanagement-Systeme (LMS) verstärken mit Selbsttests im Antwort-Auswahl-Verfahren den studentischen Eindruck, das Ziel der Physikvorlesung bestehe in der korrekten Auswahl einer von fünf Antwortmöglichkeiten. Lassen sich Selbsttest in LMS so gestalten, dass sie eine Rückmeldung über das Erreichen eines Lernziels [1] über die unterste Stufe des „Erinnerns“ hinaus liefern?

2. Didaktisches Konzept

Um mit einem Selbsttest das Erreichen von Lernzielen oberhalb des „Erinnerns“ zu überprüfen, wurde im Rahmen des Projektes die Lösungseingabe einer konventionellen Übungsaufgabe modifiziert. Nach dem hier verwendeten didaktischen Konzept „Physik durch Informatik“ [2] müssen die Studierenden die Lösung zu Übungsaufgaben in Form eines Quelltextes in der Programmiersprache Octave [3] formulieren, statt Zahlenwerte einzugeben oder eine Antwort auszuwählen. Ausgehend von konventionellen Übungsaufgaben, die häufig zur Berechnung einer physikalischen Größe auffordern („Bestimmen Sie die Eindringtiefe!“) wird die Aufgabenstellung zur Beschreibung einer Funktionsimplementierung umformuliert, („Implementieren Sie die Funktion *eindringdiefe !*“), die als Lösung eingereicht werden muss. Um die eingereichten Lösungen im LMS automatisiert zu bewerten, wird das Plug-in „CodeRunner“ [4] eingesetzt. Der von den Studierenden eingereichte Quellcode wird vom Plug-in anhand von mehreren vordefinierter Testfällen („test cases“) überprüft.

Abb.1: Beispiel einer (negativen) Rückmeldung zu einer Abgabe im Selbsttest

Die Testfälle bestehen aus Parametern zum Funktionsaufruf zusammen mit dem erwarteten Ergebnis. Der Quellcode einer Einreichung wird dann als korrekt akzeptiert, sofern er bei allen vorgesehenen Testfällen das erwartete Resultat liefert, das durch eine Musterlösung bestimmt wird.

Um zu verhindern, dass eine Einreichung ausschließlich bei den Testfällen die erwartete Lösung liefert, ist es dabei optional möglich, einzelne Testfälle vor den Studierenden zu verbergen. Zur Rückmeldung erhalten die Studierenden die Gegenüberstellung zwischen den Testfall-Ergebnissen der Musterlösung und ihrer eigenen Lösung (Abb. 1), ohne dass der Quellcode der Musterlösung angezeigt wird.

3. Pilotversuch

Das didaktische Konzept wurde in einer Physikvorlesung des ersten Semesters der beiden Bachelor-Studiengänge Biotechnologie und Umwelttechnologie im Wintersemester 2022/23 an der Hochschule Offenburg erprobt. Zusätzlich zu den zwölf Übungsblättern wurde den Teilnehmern der Vorlesung die Nutzung der wöchentlichen Selbsttests auf der LMS zur Klausurvorbereitung empfohlen. Die Studierenden wurden bei der Einschreibung in den Moodle-Kurs zufällig in zwei Gruppen A und B eingeteilt. Nach einer einführenden Aufgabe zur Nutzung der Programmier-Selbsttests, die beide Gruppen zu Übungsblatt 1 angeboten wurden, erhielten Gruppe A und B jeweils abwechselnd Selbsttests im Antwort-Auswahl-Format (Übungsblätter 2,4,6,8,10 für Gruppe A, Übungsblätter 3,5,7,9 für Gruppe B) und Programmier-Aufgaben nach dem didaktischen Konzept “Physik durch Informatik” (Übungsblätter 3,5,7,9 für Übungsgruppe A, Übungsblätter 2,4,6,8,10 für Übungsgruppe B). Die Selbsttests basierten auf Aufgabenstellungen des Arbeitsbuches zum Tipler/Mosca [3]. Das Werk wurde den Studierenden auch zur Klausurvorbereitung empfohlen. N=19 (Gruppe A) bzw. N=17 (Gruppe B) der Teilnehmenden an der Abschlussklausur Physik nahm das Angebot der freiwilligen Selbsttests mindestens einmal wahr. Bei der Abschlussklausur erreichten die beide Teilgruppen A und B ähnlich hohe Durchschnitts-Punktezahlen (48 ± 18 Punkte bzw. 48 ± 20 Punkte).

4. Ergebnisse

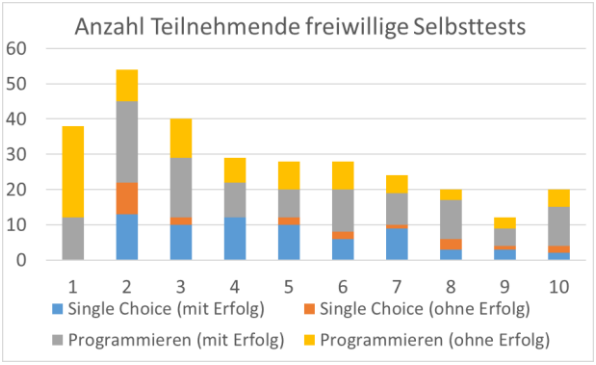

Abb.2: Teilnehmerzahlen an den freiwilligen Selbsttests

Der zeitliche Verlauf der Teilnehmendenzahlen ist in Abb.2 wiedergegeben. Am zweiten Selbsttest zu Übungsblatt 2 nahm die maximale Teilehmendenzahl (54) erreicht, davon wurden 23 als Programmierübungen (Gruppe B) und 13 Single-Choice-Aufgaben mit ähnlicher Fragestellung eingereicht. Im weiteren Semesterverlauf nahm die Teilnehmendenzahl stetig ab und erreichte beim Selbsttest zu Übungsblatt 9 mit 12 Teilnehmenden ein Minimum. Im Mittel nahmen $\mathbf{19 \pm 8}$ Teilnehmende an den Programmier-Selbsttests und etwa gleich viele ($\mathbf{15 \pm 7}$) an den Single-Choice-Selbsttests teil. Im Mittel wurden ($\mathbf{75 \pm 14}$)% der Single-Choice und ($\mathbf{64 \pm 6}$)% der Programmier-Aufgaben erfolgreich bearbeitet (Abb.3).

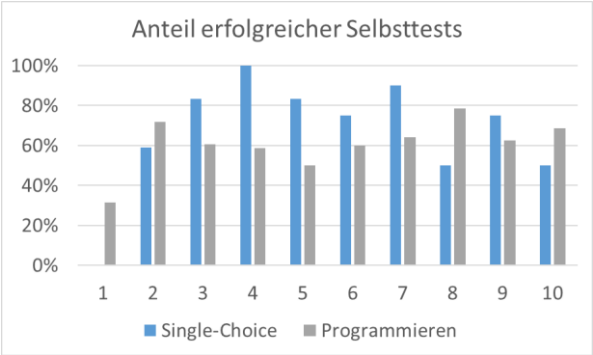

Abb.3: Anteile erfolgreicher Selbsttests (Single-Choice und Programmieren)

Zum Semesterende die Aufgabentypen im “paper-and-pen” Format evaluiert. Die Vorlesungsteilnehmer sollten dabei ihren Grad der Übereinstimmung zu jeweils 9 Aussagen zu den jeweiligen Aufgabentypen “Single-Choice” bzw. “Programmieraufgabe” angeben. Die Aussagen lauteten:

1. Manche Aufgaben vom Typ X habe ich nicht verstanden. (S1 und P1)
2. Die Aufgaben vom Typ X fand ich zu schwer. (S2 und P2)
3. Die Aufgaben vom Typ X fand ich interessant. (S3 und P3)
4. Ich hätte mir mehr Abwechslung in der Aufgabenstellung bei den Aufgaben vom Typ X gewünscht. (S4 und P4)
5. Die Aufgaben vom Typ X fand ich langweilig.(S5 und P5)
6. Der Aufgabentyp X war hilfreich für die Klausurvorbereitung. (S6 und P6)
7. Der Aufgabentyp X war hilfreich für die ein besseres Verständnis der Inhalte. (S7 und P7)
8. Der Aufgabentyp X war hilfreich für die Wiederholung des Stoffs. (S8 und P8)

9. Der Aufgabentyp X war für die Selbsteinschätzung des eigenen Wissens hilfreich. (S9 und P9)

Die Antwortmöglichkeiten und ihre Verteilung ist in Abb.4 (Aussagen 1 bis 5) und Abb.5 (Aussagen 6 bis 9) dargestellt.

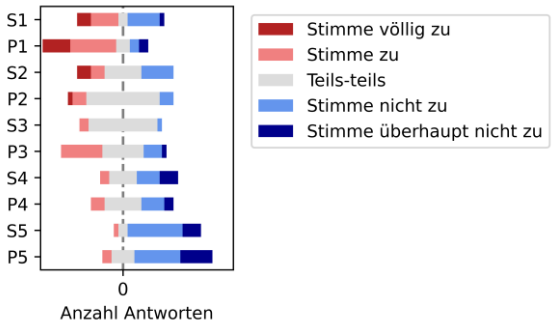

Abb.4: Verteilung der Antworten zu Aussagen S1 bis S5 zu den Single-Choice Aussagen und P1 bis P5 zu den Programmieraufgaben.

Bei den Aussagen S1 bis S5 bzw. P1 bis P6 wurden die Teilnehmer nach ihrer Einschätzung bezüglich Verständlichkeit, Schwierigkeitsgrad und Interessantheit bzw. Langeweils und Abwechslungsgrad gefragt. Hier ergibt sich die deutlichstes Abweichung bei der Verständlichkeit (S1 bzw. P1): Die Aufgabenstellungen der Programmieraufgabe wurden schlechter verstanden als die Single-Choice-Aufgaben. Ansonsten liefern Programmieraufgaben und Single-Choice-Aufgaben ähnliche Werte.

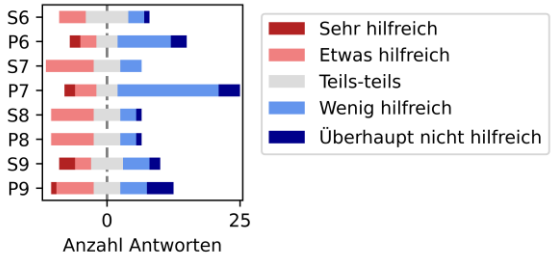

Abb. 5: Verteilung der Antworten zu Aussagen S6 bis S6 zu den Single-Choice Aussagen und P6 bis P6 zu den Programmieraufgaben.

Bei den Fragen S6 bis S9 bzw. P6 bis P9 wurden die Studierenden um ihre Einschätzung bezüglich des Nutzens („war hilfreich für“) der Aufgabentypen hinsichtlich Klausurvorbereitung, Verständnis, Wiederholung und Selbsteinschätzung gebeten. Den deutlichsten Unterschied ergibt sich bei den Fragen S7/P6, bei denen die Programmieraufgaben überwiegend als wenig hilfreich zum besseren Verständnis des Stoffes eingestuft wurden.

5. Zusammenfassung und Ausblick

Selbsttests, bei denen die Studierenden die Lösung physikalischer Aufgaben in Form eines Programmcodes eingeben müssen, werden von den Studierenden ähnlich genutzt und eingeschätzt wie vergleichbare Single-Choice-Aufgabenstellungen. Bei identischen Aufgaben schätzen die Studierenden Programmieraufgaben als schwerer verständlich ein als Fragestellungen im Antwort-Auswahl-Verfahren und als weniger hilfreicher bei der Erlangung eines besseren Verständnisses.

Die unterschiedliche Einschätzung könnte daher rühren, dass -anders als im Antwort-Auswahl-Verfahren- das Lösen der Aufgabe durch reines „Erinnern“ offensichtlich unmöglich ist, also tatsächlich eine höheres Lernziel erreicht werden muss. Um das Verständnis der Aufgabenstellung zu erleichtern, sollten daher Single-Choice-Aufgaben ergänzend eingesetz werden, um Programmieraufgaben vorzubereiten. Im direkten Vergleich zu Single-Choice-Aufgaben erscheinen Programmieraufgaben den Teilnehmern weniger hilfreich. Indem Teile der korrekten Lösung oder auch ergänzende Kommentare zur Erläuterung in das Antwortfeld voreingetragen werden, lässt sich der Aufbau des Verständnisses gezielt unterstützen. Die Teilnehmenden erarbeiten so nur kurze Passagen der Lösung wie bei einem Lückentext, um Verständnisproblemen der Studierenden zu begegnen.

6. Literatur

[1] Volk B. (2020) Ordnung von Lernzielen – Ordnung des Wissens. Die Bedeutung der Taxonomie von Bloom für die Wissenschaftlichkeit und Praxis der Hochschuldidaktik. In: Tremp P., Eugster B. (eds) Klassiker der Hochschuldidaktik. Doing Higher Education. Springer

[2] Giel, D., (2022). Physik durch Informatik. In: Henning, P. A., Striewe, M. & Wölfel, M. (Hrsg.), 20. Fachtagung Bildungstechnologien (DELFI). Bonn: Gesellschaft für Informatik e.V.. (S. 215-216). https://dx.doi.org/10.18420/delfi2022-038

[3] Eaton, J.W. Bateman, D., Hauberg, S., and Wehbring, R. (2022). GNU Octave version 7.3.0 manual: a high-level interactive language for numerical computations. https://www.gnu.org/software/octave/

[4] Lobb, R. and Harlow, J., (2016). Coderunner: A tool for assessing computer programming skills. *ACM Inroads*, *7*(1), pp.47-51.

[5] Mills, D. and Zillgitt, M. eds., 2005. Arbeitsbuch zu Tipler-Mosca Physik für Wissenschaftler und Ingenieure. Elsevier-Spektrum Akademischer Verlag.

Danksagung

Diese Arbeit wurde durch die Schwerpunktprofessur Lehrinnovation gefördert (HaW-PROAKtif FKZ 03FHP127). Ich danke Frau Daniela Schlemmer für ihre Hilfe bei der Konzeption und Auswertung der Fragebögen.