

aus

Oliver J. Bott, Peter Fricke, Uta Priss, Michael Striewe (Hrsg.)

# Automatisierte Bewertung in der Programmierausbildung

Digitale Medien in der Hochschullehre Band 6

2017, 420 Seiten, br., 42,90 €, ISBN 978-3-8309-3606-0



Waxmann Verlag GmbH

[www.waxmann.com](http://www.waxmann.com) [info@waxmann.com](mailto:info@waxmann.com)

# 1 Einleitung

Die Programmierausbildung als Bestandteil der Lehre in der Informatik und in verwandten Fächern lebt ganz wesentlich auch von praktischen Übungen anhand von Programmieraufgaben. Es ist allgemein anerkannt, dass solche Aufgaben ein wichtiger Baustein sind, um aus reinem Faktenwissen über eine Programmiersprache operationales Wissen zu machen und somit höhere Stufen in den gängigen Lernziel- oder Kompetenztaxonomien zu erreichen. Die manuelle Korrektur der von Studierenden abgegebenen Lösungen zu Programmieraufgaben ist allerdings aufwändig und erfolgt meist mit einem deutlichen zeitlichen Abstand zum Abgabezeitpunkt.

Hier liegt die Überlegung nahe, analog zu rechnergestützten Bewertungssystemen für Lösungen zu zum Beispiel Multiple-Choice- oder Mathematikaufgaben auch Lösungen zu Programmieraufgaben automatisiert bewerten zu lassen. Dies nicht nur, um insbesondere im Rahmen von E-Klausuren (respektive summativen Assessments) Zeit beziehungsweise Ressourcen für die manuelle Durchsicht zu sparen, sondern auch um den Lernenden lernunterstützend möglichst umgehend nach Abgabe der Lösung ein individualisiertes Feedback zur Korrektheit und Qualität ihrer Programme und gegebenenfalls zu Verbesserungsmöglichkeiten bereitzustellen (formatives Assessment). Der Einsatz automatisierter Bewertung für das diagnostische Assessment zum Beispiel zur Eignungsüberprüfung oder zur Einstufung in Programmierkurse ist ein weiteres mögliches Einsatzszenario. Idealerweise ermöglichen Systeme für automatisiert beurteilte Programmieraufgaben eine Darstellung des Lernfortschritts, die sowohl für Lernende als auch Lehrende hilfreich ist. Auch können Lernende durch regelmäßige und zeitnah bewertete Übungsaufgaben besser zur kontinuierlichen Mitarbeit angehalten werden. Lehrende hingegen erhalten einen Überblick der Lernfortschritte und Lernhürden einzelner Studierender und der gesamten Gruppe und können ihre Lehre daran ausrichten.

Vor diesem Hintergrund wird seit mehr als 30 Jahren intensiv an der automatisierten Bewertung von Lösungen zu Programmieraufgaben geforscht, wobei die ältesten derartigen Systeme bereits in den 1960er Jahren entwickelt und publiziert wurden. Daher verwundert es nicht, dass mittlerweile eine Vielzahl von Systemen (im Buch wird die im Englischen übliche Bezeichnung „Grader“ verwendet) mit unterschiedlichem Leistungsumfang und für verschiedene Programmiersprachen existiert. Häufig verfügen diese Grader neben der Bewertungsfunktionalität

über eine einfache Benutzer- oder Kursverwaltung. Sie bieten jedoch meist weniger Funktionalität als umfassendere Lernmanagementsysteme (LMS), welche die verschiedensten E-Learning-Bedarfe der Lehre unterstützen und mittlerweile immer häufiger eingesetzt werden. Daher ist es sinnvoll, Grader in LMS zu integrieren, wobei das LMS die Lerngruppen- und Aufgabenverwaltung und der Grader die Bewertung der von den Studierenden im LMS eingereichten Aufgabenlösungen übernimmt. Diese Integration ist allerdings mangels hierauf abgestimmter Schnittstellen zur Integration von Gradern in LMS in der Regel technisch aufwendig sowie oft auf einzelne LMS ausgerichtet und damit nicht ohne Weiteres auf andere LMS übertragbar.

Aufgrund des partiell sehr hohen Aufwands zur Entwicklung von Aufgaben für die automatisierte Programmbewertung ist die Austauschbarkeit von Aufgaben zwischen unterschiedlichen Gradern für eine Programmiersprache sowie der Aufbau von Aufgabenbibliotheken beziehungsweise Aufgaben-Repositorys ein wesentlicher Erfolgsfaktor für die automatisierte Programmbewertung. Allerdings mangelt es aktuell an international anerkannten Standards für die Beschreibung und den rechnergestützten Austausch von Programmieraufgaben, die auch für den Aufbau von LMS- beziehungsweise Grader-unabhängigen Repositorys erforderlich wären. Von Bedeutung sind zudem Algorithmen für einen effizienten Einsatz automatisierter Programmbewertung sowie die Etablierung von Konzepten zur Qualitätssicherung. Eng hiermit verbunden ist die Frage nach geeigneten didaktischen Konzepten zur Integration automatisierter Programmbewertung in die Programmierausbildung an Hochschulen und Universitäten, aber auch an Schulen und anderen Ausbildungseinrichtungen.

Zwar gibt es bereits eine Vielzahl von wissenschaftlichen Veröffentlichungen zum Thema der automatisierten Programmbewertung, unseres Wissens nach sind diese aber über verschiedene Fachcommunitys, Tagungs- und Zeitschriftenreihen verstreut. Das vorliegende Buch unternimmt den Versuch, den Stand der Technik verfügbarer Grader und deren Integration in LMS sowie von Konzepten für den didaktisch sinnvollen Einsatz automatisierter Programmbewertung nach aktuellem Kenntnisstand zusammenzustellen. Dabei fließen in dieses Buch sowohl die Erfahrungen der ersten Projektphase des eCULT-Projekts<sup>1</sup> als auch der ABP-Workshopreihe [ABP13; ABP15] ein. eCULT ist ein vom Bundesministerium für Bildung und Forschung (BMBF) gefördertes Verbundprojekt von 13 niedersächsischen Hochschulen und zwei Vereinen, das seit 2011 die Unterstützung der Präsenzlehre durch digitale Lehr- und Lerntechnologien als Mittel zur Verbesserung der Qualität der Lehre fördert. Das eCULT-Teilprojekt ProFormA<sup>2</sup>

---

1 <http://www.ecult-niedersachsen.de/>

2 ProFormA: Programming for Formative Assessment

fokussiert dabei auf Methoden zur Verbesserung der Qualität in der Programmierausbildung durch geeignete Verfahren zur automatisierten Programmbewertung. Drei der Herausgeber dieses Buches und einige der Autoren sind derzeit oder waren in vergangenen Jahren im eCULT-Projekt aktiv involviert. Insbesondere war die eCULT-ProFormA-Gruppe maßgeblich an der Entwicklung eines XML-Austauschformats für Programmieraufgaben (Kapitel 24) beteiligt. Auch die Workshops „Automatische Bewertung von Programmieraufgaben“ (ABP), die 2013 in Hannover und 2015 in Wolfenbüttel stattfanden und ein Forum für deutschsprachige Forscher zu diesem Thema boten, wurden von der eCULT-ProFormA-Gruppe mitinitiiert.

Die Ziele dieses Buches sind zum einen, Lehrenden die Anwendungsreife der automatisierten Programmbewertung durch eine geschlossene Darstellung deutlich zu machen und damit für mehr Verbreitung des Einsatzes automatischer Programmbewertung zu werben. Zum anderen soll dem Thema der automatisierten Programmbewertung ein größerer Bekanntheitsgrad in der Fachöffentlichkeit verschafft werden und auch Arbeitsergebnisse aus dem eCULT-Projekt und den ABP-Workshops einem größeren Anwenderkreis präsentiert werden. Zielgruppen des Buches sind Lehrende an Hochschulen und Schulen, Hochschuldidaktiker/innen respektive E-Learning-Serviceeinrichtungen und Informatiker/innen, die mit der Entwicklung von Gradern und Integrationslösungen befasst sind.

Das erste Kapitel des Buches bietet eine Übersicht verfügbarer Systeme zur automatisierten Programmierbewertung anhand einer Zusammenfassung von Review- und Survey-Artikeln zu diesem Thema (Kapitel 2). Das Buch schließt mit einem Ausblickkapitel, in welchem noch geplante Entwicklungen des eCULT-Projekts und offene Fragestellungen der automatisierten Programmbewertung dargestellt werden (Kapitel 25). Dazwischen ist das Buch in drei Teile gegliedert, welche im Folgenden kurz beschrieben werden.

Teil I fokussiert auf didaktische Einsatzszenarien rechnergestützter Programm-bewertung ausgehend von derzeit relevanten Programmierparadigmen und Programmiersprachen, welche in je einem Unterkapitel beschrieben sind. Im Vordergrund steht dabei nicht die Beschreibung des verwendeten Graders, sondern die Beschreibung der Einsatzszenarien mit Angaben zu Zielgruppen (z. B. Studierende eines Informatikstudiengangs im ersten Studienabschnitt), Veranstaltungen (z. B. „Einführung in die objektorientierte Programmierung“) und didaktischen Konzepten der Veranstaltungen, falls relevant ergänzt um Angaben zu den allgemein eingesetzten digitalen Medien und E-Learning-Konzepten (z. B. Blended Learning unter Einsatz eines LMS). Primäre Zielgruppe dieses Teils sind Lehrende an Hochschulen, Schulen und in der beruflichen Ausbildung, die automatisierte Programmbewertung in ihre Lehre integrieren möchten. Weiterhin sind die Unter-

kapitel interessant für Mitarbeiter von E-Learning-Serviceeinrichtungen, die ein Beratungsangebot für Werkzeuge zur automatisierten Programmbewertung aufbauen wollen.

Teil II des Buches nimmt eine technisch orientierte Sichtweise ein und betrachtet verschiedene Systeme zur Bewertung von Aufgabenlösungen. Diese Grader haben zum Teil sehr unterschiedliche Anwendungsbereiche, Schwerpunkte und Entwicklungsgeschichten. Ihre Auswahl ist exemplarisch und erhebt keinen Anspruch auf Vollständigkeit, da eine umfassende Darstellung aller aktuell und früher verfügbaren Systeme den Rahmen dieses Buches mehr als sprengen würde. Ein Kriterium für die Auswahl der betrachteten Grader ist allerdings, dass die Grader zum Zeitpunkt der Buchveröffentlichung aktiv in der Lehre eingesetzt werden und auch für Interessierte potenziell verfügbar sind. Die ausgewählten Grader werden in den jeweiligen Kapiteln detailliert vorgestellt und am Ende dieses Teils des Buches mit kurzen Tool-Steckbriefen und einer Feature-Matrix übersichtlich einander gegenübergestellt. Primäre Zielgruppen dieses Teils sind die wissenschaftlich und technisch interessierte Fachcommunity (insbesondere Entwickler von Gradern und Gradingverfahren) sowie Mitarbeiter von E-Learning-Servicestellen, die geeignete Grader für die Lehrenden ihrer Einrichtung auswählen und bereitstellen wollen. Weiterhin interessant sind die Kapitel für an spezifischen Gradern interessierte Lehrende.

In Teil III des Buches wird die Interoperabilität von Gradern untereinander und mit LMS thematisiert. In den ersten vier Unterkapiteln steht die Perspektive der LMS im Vordergrund, es werden verschiedene Ansätze und die damit verbundenen Integrationsaufwände und -probleme der verschiedenen LMS vorgestellt. Hierbei werden im deutschsprachigen Raum bekannte LMS, wie Moodle (Kapitel 18), LON-CAPA (Kapitel 19), Stud.IP (Kapitel 20) und ILIAS (Kapitel 21) anhand von standortbezogenen Erfahrungen vorgestellt. In weiteren zwei Unterkapiteln werden Middleware-Konzepte diskutiert, welche die Arbeiten und Aufwände an den Schnittstellen der LMS und Grader vereinfachen sollen. Um Programmieraufgaben zwischen LMS, aber auch Gradern austauschbar zu machen wird im letzten Kapitel das Austauschformat der ProFormA-Gruppe erläutert. Primäre Zielgruppen dieses Teils sind die wissenschaftlich und technisch an automatisierter Programmbewertung interessierte Fachcommunity sowie Mitarbeiter von E-Learning-Servicestellen, die Grader und Aufgabenbibliotheken für die Lehrenden ihrer Einrichtung integriert in das ebenda verwendete LMS bereitstellen wollen.

Einige Themen bleiben in diesem Buch bewusst ausgeklammert, ohne ihnen damit Relevanz im Rahmen der automatischen Bewertung von Programmieraufgaben absprechen zu wollen. Gleich aus zwei Gründen gibt es kein eigenes Kapi-

tel über Plagiatserkennung: Einerseits sind Plagiate kein spezielles Phänomen der Programmierausbildung, sondern sind in nahezu jedem Fach anzutreffen. Andererseits gibt es in vielen Graden sehr gute individuelle Lösungen, die im Detail in einigen Kapiteln von Teil II beschrieben werden. Das Thema ist aber auch in der Softwareentwicklung generell relevant im Hinblick auf Codeanalysen und Patentverletzungen. Eine umfassende Darstellung dieses Themas müsste also weit über den Rahmen dieses Buches hinausgehen. Gleiches gilt für den Datenschutz mit Blick auf die sensiblen, personenbezogenen Daten der Studierenden, die insbesondere im Rahmen von Prüfungen anfallen. Auch hier wird bei der Vorstellung einzelner Grader auf besonders bemerkenswerte Lösungen eingegangen, während eine umfassende Darstellung weit über die Programmierausbildung hinaus reichen müsste. Auch andere für das E-Learning relevante Themen, wie zum Beispiel Open Educational Resources (OER), Lizenzierung von Software und Aufgaben bezüglich des Austauschs und IT-sicherheitsrelevante Fragestellungen können im Rahmen dieses Buches nur randständig angesprochen werden.

Trotz dieser Auslassungen hoffen wir, Ihnen mit diesem Buch einen Überblick über den Stand der Technik sowie die Potenziale und aktuellen Grenzen des Einsatzes automatisierter Programmbewertung in der Programmierausbildung verschaffen zu können. Noch mehr hoffen wir, dass dieses Buch Lehrende ermutigt, automatisierte Programmbewertung in der Lehre einzusetzen, sowie E-Learning-Servicestellen unterstützt, geeignete Grader idealerweise integriert in das hauseigene LMS anzubieten. Und vielleicht trägt das Buch ja sogar dazu bei, dass sich weitere Informatikerinnen und Informatiker in die aktive Weiterentwicklung der automatisierten Programmbewertung einbringen.

Wir danken den zahlreichen Autoren für deren Beiträge zu diesem Buch, dem ELAN e.V. für die Aufnahme dieses Buches in dessen Reihe „Digitale Medien in der Hochschullehre“ und die damit verbundene finanzielle Förderung dieses Buchprojekts sowie dem BMBF für die Unterstützung der partiell dieser Veröffentlichung zugrunde liegenden Projekte. Weiterhin danken wir der Hochschule Hannover, der Ostfalia Hochschule für angewandte Wissenschaften, der Technischen Universität Clausthal sowie der Universität Osnabrück für die Förderung des Buchprojekts. Schlussendlich danken wir dem Waxmann Verlag für die hervorragende Betreuung des Veröffentlichungsprozesses.

Oliver J. Bott, Peter Fricke, Uta Priss und Michael Striewe  
Hannover, der 17.03.2017

**Literatur für dieses Kapitel**

- [ABP13] *Workshop „Automatische Bewertung von Programmieraufgaben“ (ABP 2013)*. Bd. 1067. CEUR Workshop Proceedings. (<http://ceur-ws.org/Vol-1067/>). 2013.
- [ABP15] *Workshop „Automatische Bewertung von Programmieraufgaben“ (ABP 2015)*. Bd. 1496. CEUR Workshop Proceedings. (<http://ceur-ws.org/Vol-1496/>). 2015.