

GATE - Ein System zur Verbesserung der Programmierausbildung und zur Unterstützung von Tutoren

Oliver Müller
Institut für Informatik
Technische Universität Clausthal
Julius-Albert-Str. 4
38678 Clausthal-Zellerfeld
oliver.mueller@tu-clausthal.de

Sven Strickroth
Institut für Informatik
Humboldt-Universität zu Berlin
Unter den Linden 6
10099 Berlin
sven@strickroth.net

1 Einleitung

Zur Verbesserung der Programmierausbildung und zur Unterstützung von Tutoren, insbesondere in Vorlesungen mit mehreren hundert Studierenden, wurde an der TU Clausthal das webbasierte, plattformunabhängige, GPLv3 lizenzierte System GATE¹ (Generic Assessment & Testing Environment) [Str09] entwickelt. Das System bietet die Möglichkeit Übungen zu organisieren und zu verwalten sowie Lösungen zu Übungsaufgaben in digitaler Form einzureichen und innerhalb des Systems zu korrigieren bzw. zu bewerten. Zusätzlich stellt es weitere spezielle Funktionen für Aufgaben aus den Bereichen Java-Programmierung und UML-Modellierung zur Verfügung.

Dieses Paper gibt zunächst einen kurzen Überblick über die wesentlichen Funktionen, die GATE Studierenden und Tutoren bietet, und geht anschließend auf die für das System durchgeführten Evaluationen ein, die bereits in ausführlicherer Form in [SOP11] bzw. [Sch+12] veröffentlicht wurden.

2 Funktionen des Systems GATE

2.1 Funktionen für Studierende

Studierende, die an einer Vorlesung teilnehmen, können sich mit GATE in Übungsgruppen eintragen, ihre aktuellen Punktestände einsehen und ihre Lösungen zu Übungsaufgaben ohne Medienbruch direkt in digitaler Form einreichen. Sind Testfälle freigegeben, können Studierende begrenzt oft automatisches Feedback für ihre Lösung anfordern und innerhalb des Bearbeitungszeitraumes eine modifizierte Lösung einreichen. Durch das Feedback soll die Qualität der studentischen Abgaben weiter verbessert werden.

2.2 Funktionen für Tutoren

Eingereichte Lösungen sind nach Abgabeschluss einer Aufgabe von den Tutoren einsehbar und können dann mit Hilfe des Systems korrigiert und bewertet werden. Über eine Freitext-Kommentarfunktion können Tutoren ausführlicheres Feedback zu ihren Bewertungen und Korrekturen bzw. zu den studentischen Lösungen geben. Das Feedback kann hierbei nicht nur an die jeweils betroffenen Studierenden, sondern auch an alle anderen Tutoren adressiert sein, die derselben Veranstaltung zugeteilt sind und jeweils eine andere Übungsgruppe betreuen. Hierdurch können sich die Tutoren untereinander z.B. über

¹Link zum Quellcode: <https://github.com/csware/si>

vermutete oder bestätigte Plagiate informieren oder eine weitere Korrektur durch einen anderen Tutor anfordern.

Um den insbesondere bei großen Vorlesungen anfallenden manuellen Korrektur- bzw. Bewertungsaufwand bezüglich eingereichter Lösungen zu reduzieren, stellt GATE den Tutoren Ergebnisse automatisch ausgeführter Tests sowie eine automatische Plagiatserkennung zur Verfügung. Für das System stehen drei Plagiat-Erkennungsalgorithmen zur Verfügung: der Plagie-Algorithmus [ASR06], die Berechnung der Levenshtein-Distanz [Lev66] und die Berechnung der Normalized-Compression-Distance (NCD) [Li+04]. Für die Plagiatsprüfung lassen sich die Abgaben normalisieren. Es kann somit erreicht werden, dass z.B. Kommentare im Quellcode oder Leerzeichen bei einem Prüfvorgang nicht beachtet werden.

2.3 Neueste Entwicklung

Die neueste Entwicklung für das System GATE sind randomisierte Aufgaben: In den Aufgabenstellungen können Variablen benutzt werden, die für jeden Studierenden mit individuellen Werten belegt werden. Studierende werden gebeten das Endergebnis in separate Textboxen sowie den Rechenweg in eine weitere Textbox einzugeben. Endergebnisse lassen sich hierbei vollständig automatisch überprüfen. Tutoren erhalten für die Bewertung sämtliche Variablen Belegungen und bekommen Lösungen für Zwischenergebnisse sowie Lösungen für Endergebnisse angezeigt, damit sie Fehler schnell finden und korrigieren können. Einige andere Systeme wie z.B. LON-CAPA² zeigen zwar End- nicht aber Zwischenergebnisse an.

Ein weiterer Vorteil solcher Aufgaben ist, dass Studierende über Aufgaben diskutieren können, aber jeder die Aufgabe selber lösen muss. Da jeder Studierende bei diesem Aufgabentyp die gleiche Aufgabe, aber mit individuellen Werten erhält, traten Plagiate in den bisherigen Veranstaltungen, bei denen GATE eingesetzt wurde, bezüglich randomisierter Aufgaben nicht auf.

Die Tabelle auf der folgenden Seite gibt für die angegebenen randomisierten Aufgaben die im Schnitt erreichten Punkte an. Die Daten wurden dabei aus dem System GATE entnommen. Die Aufgaben mussten im Rahmen der an der TU Clausthal stattfindenden Veranstaltung "Grundlagen der Programmierung" für BWL Studenten im WS 2011/2012 bzw. im SS 2013 gelöst werden. Bei der Veranstaltung im SS 2013 handelte es sich um einen kleinen Kurs, bei dem die Möglichkeit bestand einen Übungsschein auch ohne Vorlesung zu erlangen. Hierdurch erklärt sich auch die geringe Anzahl an Abgaben für die beiden angegebenen Aufgaben aus dem SS 2013. Zum besseren Verständnis erfolgt an dieser Stelle eine kurze Beschreibung der in der Tabelle aufgeführten Aufgabentypen.

- Speicherkalkulation: Verfügbaren Speicherplatz von Speichermedien berechnen
- Fließkommazahlen: Verschiedene Darstellungsformen von Fließkommazahlen bestimmen (Hexadezimaldarstellung, Dezimaldarstellung, Binärfolge)
- Videokalkulation: Speicherbedarf eines Videos berechnen
- Komplexe Datentypen: Speicherbedarf komplexer Datentypen berechnen

²<http://www.lon-capa.org>

Aufgabentyp	Abgenommene Abgaben	Max. Punkte	ØPunkte
Speicheralkulation (WS 11/12)	313	10	8,18 (81,8%)
Fließkommazahlen (WS 11/12)	302	10	7,46 (74,6%)
Videokalkulation (WS 11/12)	306	6	5,29 (88,2%)
Komplexe Datentypen (WS 11/12)	295	6	4,18 (69,6%)
Speicheralkulation (SS 13)	13	10	9 (90%)
Fließkommazahlen (SS 13)	13	10	5,76 (57,6%)

Für Aufgaben vom Typ Speicheralkulation waren bei der gleichen Veranstaltung im WS 2009/2010, in denen keine randomisierten Aufgaben zum Einsatz kamen, im Schnitt ähnliche Punktzahlen zu verzeichnen (Abgenommene Abgaben: 242, Max. Punkte: 10, ØPunkte: 8,19 (81,9%)). Bei nicht randomisierten Aufgaben vom Typ Fließkommazahlen war bei der Veranstaltung im WS 2009/2010 (Abgenommene Abgaben: 230, Max. Punkte: 10, ØPunkte: 5,74 (57,4%)) die durchschnittlich erreichte Punktzahl schlechter als bei der Veranstaltung im WS 2011/2012. Im WS 2010/2011 (Abgenommene Abgaben: 224, Max. Punkte: 10, ØPunkte: 7,24 (72,4%)) war sie wiederum ähnlich hoch wie im WS 2011/2012. Ähnliche Aufgaben vom Typ Videokalkulation bzw. Komplexe Datentypen wurden im WS 2009/2010 bzw. im WS 2010/2011 nicht gestellt.

Insgesamt lässt sich feststellen, dass in den Veranstaltungen, bei denen randomisierte Aufgaben zum Einsatz kamen (WS 2011/2012, SS 2013), im Schnitt recht hohe Punktzahlen erreicht wurden (Ausnahme: Aufgabe vom Typ Fließkommazahlen im SS 2013), vergleichbar mit den durchschnittlich erzielten Punktzahlen aus den Veranstaltungen der vorangegangenen Semester, in denen keine randomisierten Aufgaben gestellt wurden.

2.4 Zusammenfassung

Das System GATE ist also nicht nur ein reines Online Abgabesystem, das verwaltungs- bzw. organisationstechnische Aufgaben des Übungsbetriebs unterstützt. Es integriert darüber hinaus weitere Funktionen, um die Arbeit von Tutoren zu erleichtern bzw. Studierende bei der Erstellung ihrer Lösungen zu unterstützen. Bei diesen Funktionen handelt es sich u.a. um Syntax- und Funktionstests für Java-Programmieraufgaben bzw. spezielle Tests für UML-Aufgaben, die sowohl Tutoren als auch begrenzt oft Studierenden zur Verfügung stehen, sowie um eine automatische Plagiatsprüfung, die nur von Tutoren genutzt werden kann.

3 Evaluation des Systems GATE

3.1 Evaluation zum Einsatz bei Java-Programmieraufgaben

GATE wurde erstmalig an der TU Clausthal im Rahmen der Vorlesung "Grundlagen der Programmierung" für BWL-Studenten 2009 eingesetzt und evaluiert [SOP11]. Den Studierenden sollte in dieser Vorlesung Basiswissen zu Datentypen, UML und v.a. Programmieren vermittelt werden. Als Programmiersprache wurde Java verwendet. Für ein erfolgreiches Absolvieren der Vorlesung wurden die Studierenden dazu aufgefordert, maximal in Zweiergruppen regelmäßig und selbständig Aufgaben zu bearbeiten und schließlich einem Tutor

persönlich zu präsentieren. Für die Evaluation waren insbesondere die folgenden Fragestellungen in Bezug auf die zu bearbeitenden Java-Programmieraufgaben relevant.

- Hilft GATE den Tutoren bei der Bewertung von eingesandten Lösungen sowie bei der Erkennung von Plagiaten?
- Wie werden die Tests von Studierenden genutzt?
- Steigern die für die Studierenden verfügbaren Tests die Qualität der abgegebenen Lösungen?

3.1.1 Ergebnisse der Evaluation

Bei der Evaluation konnte beobachtet werden, dass eine Qualitätssteigerung studentischer Lösungen durch das Konzept der studentischen Tests möglich ist. Dies spiegelte sich in der Tatsache wider, dass zum einen im Schnitt die Lösungen derjenigen Studierenden, die einen Syntaxtest durchführten, zu 90% syntaktisch korrekt waren, Lösungen von Studierenden, die keinen Syntaxtest nutzten, im Schnitt dagegen nur zu 65%. Die Differenz von 25% stellte sich dabei als statistisch signifikant heraus. Ein ähnliches Bild zeigte sich bezüglich der Funktionstests. Hierbei waren die Abgaben, die einem Funktionstest unterzogen wurden, durchschnittlich zu 52% korrekt und die Abgaben, für die kein Funktionstest durchgeführt wurde, im Schnitt nur zu 18%. Auch bei dieser Differenz von 34% konnte eine statistische Signifikanz festgestellt werden.

Die automatischen Plagiatsprüfungen erleichterten es den Tutoren, Plagiate unter den eingesandten Lösungen zu finden. Bei Programmieraufgaben wurde diesbezüglich der Plagie-Algorithmus von den Tutoren als sehr nützlich empfunden (durchschnittliche Bewertung 4,3 auf einer Skala von 1 bis 5) die Levenshtein-Distanz und die NCD dagegen weniger (durchschnittliche Bewertung 2,8 Punkte auf einer Skala von 1 bis 5).

Nach Angabe der Tutoren reduzieren die durch GATE zur Verfügung gestellten Funktionen den Korrektur- und Bewertungsaufwand. Hierbei verließen sich die Tutoren allerdings nicht alleine auf die durch das System zur Verfügung gestellten Hilfsmittel, sondern nahmen weiterhin Bewertungen mit Augenmaß vor. Dies lag vor allem auch an den Funktionstests, die mehrere studentische Lösungen als falsch negativ klassifizierten. Von 1031 Lösungen wurden durch die Funktionstests 759 Lösungen (73,6%) als nicht korrekt klassifiziert aber 201 Lösungen (26,5%) hiervon von den Tutoren trotzdem als korrekt bewertet, da nur kleinere Fehler vorhanden waren (z.B. Tippfehler, Package falsch gesetzt).

Das System GATE wurde seit dieser Evaluation in den folgenden Veranstaltungen an der TU Clausthal eingesetzt, bei denen die jeweils in Klammern angegebenen Aufgabentypen behandelt wurden.

- “Grundlagen der Programmierung” für BWL-Studenten (Java-Programmieraufgaben, Textaufgaben, Aufgaben mit randomisierten Werten, UML-Aufgaben)
- “Programmierkurs” für Informatik-/Wirtschaftsinformatik-Studenten (Java-Programmieraufgaben)
- “Informatik I” (Textaufgaben, C-Programmieraufgaben (nur Abgabe und manuelle Bewertung/Korrektur))

- “Rechnerübung Unternehmensforschung I” (Java-Programmieraufgaben)
- “Mensch-Maschine-Kommunikation” (Textaufgaben)

3.2 Studie zum Einsatz bei UML-Modellierungsaufgaben

Nach den positiven Ergebnissen bei Java-Programmieraufgaben wurde im Rahmen einer Studie mit 30 Personen, die über keine oder über nur wenige Programmier- bzw. Modellierungskennntnisse verfügen, untersucht, ob sich ähnliche Ergebnisse auch für UML-Aufgaben, speziell bei Klassen- und Aktivitätsdiagrammen, erreichen lassen. GATE wurde dazu um einen grafischen UML-Editor (basierend auf ArgoUML³) sowie spezielle Feedbackoptionen für UML-Aufgaben erweitert [Sch+12]. In der Studie wurde untersucht, ob es möglich ist, mit dem erweiterten System Modellierungsfähigkeiten besser zu erlernen als durch den Einsatz eines reinen UML-Tools ohne Feedback Funktionen bzw. in Kombination mit einem Tutor, der für Feedback zur Verfügung steht.

3.2.1 Ergebnisse der Studie

Eine Befragung der Studienteilnehmer, die das erweiterte GATE-System nutzten, ergab, dass über das System ein guter Lerneffekt erzielt werden kann (Bewertung auf einer Skala von 0 bis 5 im Schnitt 4,4) und zudem die UML spezifischen Feedbackoptionen als sehr nützlich empfunden werden (Bewertung auf einer Skala von 0 bis 5 im Schnitt 4,3).

Qualitativ waren die Lösungen, die über das erweiterte GATE-System erstellt wurden (im Schnitt 94,4 von 104 Punkten) besser als die Lösungen, die nur mit Hilfe von ArgoUML entstanden (im Schnitt 82,2 von 104 Punkten). Die Studienteilnehmer, denen ein Tutor für Feedback zur Verfügung stand, schnitten in diesem Zusammenhang am besten ab (im Schnitt 100,9 von 104 Punkten). Die Hypothese, dass durch den Einsatz des erweiterten GATE-Systems Modellierungsfähigkeiten statistisch signifikant besser erlernt werden können als durch die Verwendung eines UML Tools ohne Feedbackfunktion, konnte dabei nicht bestätigt werden. Ebenso ergab sich kein statistisch signifikanter Unterschied zur Qualität der Lösungen der Gruppe, die ihre Aufgaben mit Hilfe eines Tutors bearbeiteten.

4 Zusammenfassung und Ausblick

Das Paper stellt das webbasierte, plattformunabhängige System GATE vor. Bei diesem handelt es sich um ein Online Abgabesystem zur Verwaltung bzw. Organisation des Übungsbetriebs an Universitäten. Zur Reduzierung des Bewertungs- und Korrekturaufwands für Tutoren und zur Verbesserung der Qualität studentischer Lösungen zu Übungsaufgaben, sind darüber hinaus weitere Funktionen integriert. Neben einer automatischen Plagiatsprüfung, die ausschließlich von Tutoren genutzt werden kann, stehen sowohl Tutoren als auch Studierenden Syntax- und Funktionstests für Java-Programmieraufgaben und spezielle Tests für UML Aufgaben zur Verfügung.

Eine Evaluation im Rahmen einer Programmiervorlesung an der TU Clausthal, bei der als Programmiersprache Java verwendet wurde, ergab, dass der Einsatz von GATE sowohl für

³<http://argouml.tigris.org/>

Studierende als auch für Tutoren einen Mehrwert bietet. Zum einen ließ sich eine Qualitätssteigerung der studentischen Lösungen durch den Einsatz der studentischen Tests erreichen und zum anderen konnten Plagiate durch die automatische Plagiatsprüfung leichter erkannt werden. Insgesamt ließ sich der Korrektur- bzw. Bewertungsaufwand für die Tutoren durch den Einsatz der Funktionen des Systems GATE reduzieren.

Ein um einen graphischen UML Editor und spezielle Feedbackfunktionen erweitertes GATE-System wurde in einer weiteren Studie evaluiert. Die aufgestellte Hypothese (Mit dem erweiterten GATE-System lassen sich Modellierungsfähigkeiten besser erlernen, als mit einem UML Tool ohne Feedbackfunktionen) konnte hierbei nicht bestätigt werden. Dennoch wurde das erweiterte GATE-System von den Studienteilnehmern sehr positiv bewertet.

Da GATE eine modulare Architektur besitzt, ist es denkbar weitere Tests zu integrieren, um systemseitiges Feedback zu weiteren Programmier- bzw. Modellierungssprachen erhalten zu können. Ebenso bietet es sich an die Plagiatsprüfung z.B. durch das Einbinden weiterer Plagiatsalgorithmen zu optimieren, um diesbezüglich noch bessere Ergebnisse zu erzielen. In Bezug auf die Plagiaterkennung wäre es zudem von Vorteil, wenn studentische Lösungen mit ähnlichen Lösungen aus dem Internet verglichen werden könnten.

Literatur

- [ASR06] Aleksi Ahtiainen, Sami Surakka und Mikko Rahikainen. „Plaggie: GNU-licensed source code plagiarism detection engine for Java exercises“. In: *Proceedings of the 6th Baltic Sea conference on Computing education research: Koli Calling 2006*. Baltic Sea '06. Uppsala, Sweden: ACM, 2006, S. 141–142. DOI: 10.1145/1315803.1315831. URL: <http://doi.acm.org/10.1145/1315803.1315831>.
- [Lev66] Vladimir I. Levenshtein. „Binary Codes Capable of Correcting Deletions, Insertions and Reversals“. In: *Soviet Physics Doklady* 10 (1966), S. 707.
- [Li+04] Ming Li u. a. „The similarity metric“. In: *Information Theory, IEEE Transactions on* 50.12 (2004), S. 3250–3264.
- [Sch+12] J. Schramm, S. Strickroth, N. T. Le und N. Pinkwart. „Teaching UML Skills to Novice Programmers Using a Sample Solution Based Intelligent Tutoring System“. In: *Proceedings of the 25th International Conference of the Florida Artificial Intelligence Research Society (FLAIRS)*. Hrsg. von G. M. Youngblood und P. McCarthy. Marco Island, FL, USA: AAAI, 2012, S. 472–477.
- [SOP11] S. Strickroth, H. Olivier und N. Pinkwart. „Das GATE-System: Qualitätssteigerung durch Selbsttests für Studenten bei der Onlineabgabe von Übungsaufgaben?“. In: *Tagungsband der 9. e-Learning Fachtagung Informatik (DeLFI)*. Hrsg. von H. Rohland, A. Kienle und S. Friedrich. Bd. P-188. GI Lecture Notes in Informatics P-188. Bonn, Germany: GI, 2011, S. 115–126. ISBN: 978-3-88579-282-6.
- [Str09] S. Strickroth. „Unterstützungsverfahren für Tutoren bei der Online-Abgabe von Übungsaufgaben“. Bachelor Thesis. Clausthal University of Technology, Department of Informatics, 2009.