

# Werkzeuggestützte Untersuchung der Vorgehensweisen von Lernenden beim Lösen algorithmischer Probleme

Ulrich Kiesmüller, Torsten Brinda

Didaktik der Informatik  
Friedrich-Alexander-Universität Erlangen-Nürnberg  
Martensstraße 3  
91058 Erlangen  
{ulrich.kiesmueller, torsten.brinda}@informatik.uni-erlangen.de

**Abstract:** Um Lernende der Sek. I beim Lösen algorithmischer Probleme unter Verwendung von Lern- und Programmierumgebungen besser zu unterstützen, wird untersucht, inwieweit deren Vorgehensweisen automatisch erfasst und bewertet werden können, mit dem Ziel die eingesetzten Umgebungen lernergerichteter zu gestalten. Dazu wurde für die Umgebung Kara Aufzeichnungs- und Diagnosesoftware entwickelt und in empirischen Vorstudien auf ihre Praxistauglichkeit hin untersucht.

## 1 Motivation und Ziel

Die in der Sekundarstufe I in der Informatikanfangsausbildung eingesetzten Systeme zum Erlernen algorithmischer Grundkonzepte (z. B. Karol, Kara) geben den Lernenden nicht an deren individuelle Vorgehensweisen angepasste, oft rein „technische“, wenig schülergeeignete Fehlermeldungen aus. Die Hypothese für das hier skizzierte Forschungsprojekt lautet, dass sich die eingesetzten Lern- und Programmierumgebungen bei genauerer Kenntnis der individuellen Vorgehensweisen bei der Bearbeitung algorithmischer Problemstellungen besser an die Bedürfnisse der Lernenden anpassen lassen. Untersuchungen zu den Vorgehensweisen von Programmieranfängern fanden bislang überwiegend auf Hochschulebene statt (z. B. [Hu06]). Hierbei wurde häufig mit Videoaufzeichnungen gearbeitet, die nachträglich aufwändig manuell ausgewertet werden mussten, um das jeweilige Vorgehen zu analysieren. Um sowohl zu quantitativen als auch zu qualitativen Aussagen zu gelangen, werden im Rahmen der hier vorgestellten Studie einerseits Aufzeichnungs- und Diagnosesoftware konzipiert und entwickelt und andererseits Einzelinterviews mit Lernenden zu ihren spezifischen Vorgehensweisen durchgeführt. Angestrebt wird, automatisch diagnostizierte Vorgehensweisen von Lernenden für die Gestaltung von individualisierten Systemrückmeldungen zu verwenden.

## 2 Entwicklung von Aufzeichnungs- und Diagnosesoftware

Grundlage für die ersten Arbeiten bildete die Lernumgebung Kara [Re03]. Für die automatisierte Erfassung quantitativer Daten wurde in einem ersten Schritt eine Aufzeich-

nungssoftware (TrackingKara) entworfen und implementiert, die den zeitlichen Verlauf aller relevanten Aktionen eines Lernenden für diesen unsichtbar in einer Textdatei protokolliert. Hierzu wurde vom Kara-Team (insbesondere R. Reichert) eine spezifische Programmschnittstelle bereitgestellt, wofür die Autoren herzlich danken. Aufgezeichnet werden alle Aktionen, die die Weiterentwicklung und die Ausführungsversuche eines Kara-Programms durch den Lernenden dokumentieren. Endet ein Ausführungsversuch eines Kara-Programms mit einer Fehlermeldung, so wird auch diese protokolliert zusammen mit der Information, wie die Bearbeitung anschließend fortgesetzt wird. Um einerseits die Vorgehensprotokolle einzelner Lernender grafisch aufbereitet (z. B. relevante Ausbaustufen des Kara-Programms, Aktionen im Zeitverlauf) analysieren zu können und um andererseits kumulative Aussagen über die Verläufe mehrerer Probanden treffen zu können, wurde eine zusätzliche Diagnosesoftware (EvalKara) entwickelt und implementiert, in der die Protokollinformationen in einer Datenbank verwaltet werden.

### **3 Erste Untersuchungen, Ergebnisse und Ausblick**

Im 1. Halbjahr 2007 wurde die Praxistauglichkeit der entwickelten Analyse- und Diagnosesoftware in Fallstudien untersucht. Hierbei nahmen in Einzelfallstudien ca. 10 Personen mit sehr unterschiedlicher informatischer Vorbildung (keine bis erhebliche) sowie ca. 100 Lernende aus der Jgst. 7 des bayerischen Gymnasiums teil. Hierbei zeigte sich, dass durch die Aufzeichnung der zeitlichen Veränderung der Anzahl der Programmelemente sowie der gewählten Aktionen Schlussfolgerungen im Hinblick auf Vorgehensmuster (z. B. evolutionäres/Versuch-Irrtum-Verfahren, strategisches Vorgehen) möglich sind. Erkenn- und bewertbar ist weiterhin, wie viel Zeit ein Lernender mit welchen Aktivitäten (Editieren von Zuständen und Übergängen, der Modellwelt, Ausführen) verbringt sowie Fehlerart und -häufigkeit pro untersuchter Aufgabenstellung. Die Qualität des erreichten Bearbeitungsergebnisses eines Lernenden wird durch systemunterstützte Überprüfung der Lösung für vorgegebene Testdaten erfasst. Hier ist im Weiteren noch ein Abgleich mit den Erkenntnissen aus dem Bereich der automatischen Überprüfung von Programmieraufgaben geplant.

In weiteren Arbeiten soll die Erfassung der Vorgehensweisen verfeinert und mit Empfehlungen für die Lernenden verknüpft werden. Hierzu ist auch der Einsatz weiterer Lern- und Programmierumgebungen geplant. Die Auswirkung der Einbindung der Empfehlungen in die Software auf die Lernprozesse ist Gegenstand weiterer empirischer Arbeiten.

### **Literaturverzeichnis**

- [Hu06] Hundhausen, C. D.: A Methodology for Analyzing the Temporal Evolution of Novice Programs Based on Semantic Components. In: Proceedings of ICER '06, ACM Press, New York, 2006, pp. 59-71.
- [Re03] Reichert, R.: Theory of Computation as a Vehicle for Teaching Fundamental Concepts of Computer Science. Dissertation 15035, ETH Zürich, 2003.