

**Prozessbegleitende automatisierte Identifizierung  
von Problemlösestrategien Lernender  
beim Lösen algorithmischer Probleme mit  
Programmierungsumgebungen in der Sekundarstufe I**



**Der Technischen Fakultät der  
Universität Erlangen-Nürnberg  
zur Erlangung des Grades**

**DOKTOR-INGENIEUR**

**vorgelegt von**

**Oberstudienrat Ulrich Kiesmüller**

**Erlangen – 2012**

**Als Dissertation genehmigt von  
der Technischen Fakultät der  
Universität Erlangen-Nürnberg**

Tag der Einreichung: 08. Juni 2012  
Tag der Promotion: 28. September 2012  
Dekanin: Professorin Dr.-Ing. Marion Merklein  
1. Berichterstatter: Prof. Dr. Torsten Brinda, Erlangen  
2. Berichterstatter: Prof. Dr. Michael Fothe, Jena

Berichte aus der Informatik

**Ulrich Kiesmüller**

**Prozessbegleitende automatisierte Identifizierung  
von Problemlösestrategien Lernender  
beim Lösen algorithmischer Probleme mit  
Programmierungsumgebungen in der Sekundarstufe I**

D 29 (Diss. Universität Erlangen-Nürnberg)

Shaker Verlag  
Aachen 2013

**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.

Zugl.: Erlangen-Nürnberg, Univ., Diss., 2012

Copyright Shaker Verlag 2013

Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen oder vollständigen Wiedergabe, der Speicherung in Datenverarbeitungsanlagen und der Übersetzung, vorbehalten.

Printed in Germany.

ISBN 978-3-8440-1768-7

ISSN 0945-0807

Shaker Verlag GmbH • Postfach 101818 • 52018 Aachen  
Telefon: 02407 / 95 96 - 0 • Telefax: 02407 / 95 96 - 9  
Internet: [www.shaker.de](http://www.shaker.de) • E-Mail: [info@shaker.de](mailto:info@shaker.de)

## Kurzfassung

Lehrende der Informatik werden im Teilbereich der Algorithmik und Programmierlehre immer wieder mit Fragen konfrontiert wie „Auf welche Weise sind die Lernenden zu ihren Lösungsansätzen gelangt?“ oder „Was haben sich die Programmieranfänger bei ihren letzten Implementierungsschritten gedacht?“. Eine Beantwortung derartiger Fragen kann für Lehrende hilfreich sein, um die Lernenden möglichst effektiv bei ihrem weiteren Lösungsweg zu unterstützen oder ihre Lösungsansätze differenzierter zu bewerten als lediglich mit „völlig korrekt“ bzw. „falsch“.

Bisherige Studien dazu setzen bei der Suche nach Antworten auf obige Fragen meist Methoden ein, die lediglich eine nachträgliche Analyse der Vorgehensweisen der Lernenden mit hohem Zeitaufwand ermöglichen.

In der vorliegenden Arbeit wird ein Verfahren entwickelt, mit dessen Hilfe eine prozessbegleitende, automatisierte Identifizierung der Problemlösestrategien Lernender durchführbar ist, wenn diese algorithmische Aufgabenstellungen mittels der Programmierumgebung „Kara, der programmierbare Marienkäfer“ (vgl. Reichert [2003]) bearbeiten. Grundlage hierfür bilden Methoden und Algorithmen aus dem Bereich der Mustererkennung, die in der automatischen Spracherkennung bereits lange erfolgreich eingesetzt werden. Diese werden angepasst auf die hier betrachtete Problemstellung und als eigenes Modul in die Programmierumgebung integriert. Eine Studie zur Validierung der Ergebnisse des entwickelten Identifizierungsmoduls ergab eine sehr hohe Übereinstimmung der automatisiert identifizierten Problemlösestrategien mit den Ergebnissen menschlicher Beobachter. Auf den Resultaten des Untersuchungs- und Analysewerkzeugs basiert die Gestaltung von individualisierten Systemrückmeldungen, durch welche die Lernenden adaptiert an ihre jeweilige Vorgehensweise bei der weiteren Lösung unterstützt werden. Hierbei werden auch Aspekte der Attributionsforschung berücksichtigt, um durch die Feedbackgestaltung die Lernenden bestmöglich bei ihrer Problemlösung zu motivieren.

Bei der Entwicklung der Verfahren und Werkzeuge wurde darauf geachtet, eine gute Übertragbarkeit auf viele Programmierumgebungen zu gewährleisten. Selbst eine Anwendung in Lernumgebungen außerhalb der Informatik ist denkbar. Außerdem lassen sie sich für weitere Studien einsetzen, welche die Problemlösestrategien Lernender unter weiteren Aspekten wie z. B. Abhängigkeit der bevorzugt eingesetzten Strategien von Faktoren wie Alter, Geschlecht, Leistungsfähigkeit in anderen Fächern untersuchen.



## Abstract

Computer Science tutors are frequently confronted with questions like “What way did learners use to reach their solution approaches?” or “What did programming novices think while taking their last implementation steps?”. Answering such questions will help to support learners on their further way to a solution or to make a more differentiated evaluation of their solution approaches than just stating “completely correct” respectively “wrong”.

Searching answers to these questions previous studies in this field apply methods which only enable a subsequent analysis of the learner’s solution process at a significant cost of time.

In this work a procedure is developed by which an automated online identification of learners’ problem solving strategies is realizable when they complete algorithmic tasks using the programming environment “Kara, the programmable ladybug” (see Reichert [2003]). The basis for this is provided by methods and algorithms out of the field of pattern recognition, which have been applied successfully in automatic speech recognition for a long time. These are adapted to the problem set here and embedded in the programming environment as a separate module. A study validating the results of the developed identification tool shows very high agreement with the findings of human observers. Based on the results of this analyzing tool individualized system feedback is designed adapted to the learners’ problem solving strategy, which supports learners on their further way to a solution. With regard to aspects of the attribution theory the feedback messages are optimized for maintaining learners’ motivation during their whole problem solving process.

While developing the methods and tools a high transferability to other programming environments was ensured as far as possible. What is also easily conceivable is applying these methods with programming environments out of the area of computer science. In addition the developed procedure is employable in further studies to analyze the dependency of learners’ preferred problem solving strategies on factors like their age, their gender or their performance in other subjects.



## Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter im Rahmen meiner Vollzeitabordnung vom Gymnasium an die Universität in der Fachgruppe „Didaktik der Informatik“ der Friedrich-Alexander-Universität Erlangen-Nürnberg.

An dieser Stelle bedanke ich mich bei all denjenigen, die in unterschiedlicher Art und Weise zum Entstehen dieser Arbeit beigetragen haben. Die Arbeit wurde betreut von Herrn Prof. Dr. Torsten Brinda, dem ich herzlich danke für die interessante Themenstellung, die mir von ihm gebotenen vielfältigen nationalen und internationalen Vernetzungsmöglichkeiten sowie die intensiven Gespräche mit ihm, die mich stets weiter voranbrachten in meiner Forschungsarbeit. Bei Herrn Prof. Dr. Michael Fothe bedanke ich mich sehr für die Übernahme des Koreferats.

Großer Dank gebührt dem Lehrstuhl für Mustererkennung der Friedrich-Alexander-Universität Erlangen-Nürnberg für die zielführende und gewinnbringende Kooperation. Besonders hervorzuheben ist hierbei Korbinian Riedhammer, der mir bei Fragen und Problemen im Bereich der Mustererkennung stets mit Rat und Tat zur Seite stand.

Weiterhin danke ich den Studierenden, deren Diplomarbeit bzw. schriftliche Hausarbeiten ich betreuen durfte. Hier gilt mein Dank Frau Jasmina Žlender sowie den Herren Hannes Forster und Sebastian Sossalla. Letzterer sei insbesondere auch bedankt für die intensive Unterstützung bei der Implementierung des Softwaresystems. Den Kolleginnen und Kollegen der Fachgruppe in Erlangen danke ich für ein stets angenehmes Arbeitsklima, vielfältige Gespräche sowie viele wertvolle Hinweise im forschungsmethodischen und fachdidaktischen Bereich. Ein besonderes Danke schön verdient hier die Kollegin Simone Opel für wertvolle Hinweise beim gründlichen Korrekturlesen der Arbeit.

Schließlich geht mein Dank an die Schulen, und dort insbesondere an die Informatiklehrkräfte, die bei den Studien im Laufe dieser Arbeit mitwirkten und mich tatkräftig unterstützen – besonders zu erwähnen sind in diesem Zusammenhang Herr StR Harald Lamprecht, Frau StRin Sandra Leibinger und Herr OStR Herbert Steinmetz.

Des Weiteren bedanke ich mich hier bei allen nationalen und internationalen Kollegen, die mir mit intensiven Gesprächen eine wertvolle Hilfestellung für die Weiterentwicklung und Fertigstellung dieser Arbeit boten. Stellvertretend seien hier genannt Prof. Dr. Tony Clear, Dr. Maria Knobelsdorf, Prof. Dr. Lauri Malmi und Dr. Ralf Romeike, mit denen über die Dauer meiner Forschungsarbeit hinweg ein besonders intensiver Gedankenaustausch stattfand.

Hier zwar an letzter Stelle genannt, aber für mich sicher die größte und wichtigste Unterstützung, war die durch meine Familie. Herzlich bedanke ich mich bei meinen

Söhnen Stefan und Johannes und insbesondere bei meiner Frau Brigitte Gottwald, die in den Jahren meiner universitären Tätigkeit zahlreiche Entbehrungen auf sich nahmen und mich mit übergroßer Geduld stets sowohl im persönlichen Bereich als auch in fachlicher Hinsicht mit all ihren Kräften unterstützten. Ein großer Dank geht an meine Schwiegereltern Annemarie und Bruno Gottwald, die mich immer wieder zum Durchhalten ermunterten. Auch meinen Eltern Christa und Peter danke ich für die gewinnbringenden Diskussionen über meine Arbeit und das sorgfältige Korrekturlesen der Arbeit.

Erlangen, im Juni 2012

# Inhaltsverzeichnis

<b>Kurzfassung</b>	<b>III</b>
<b>Abstract</b>	<b>V</b>
<b>Vorwort</b>	<b>VII</b>
<b>Inhaltsverzeichnis</b>	<b>IX</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Vermittlung der Algorithmik im Unterricht . . . . .	1
1.2 Erfahrungen aus dem Informatikunterricht . . . . .	2
1.3 Problemstellung . . . . .	3
1.4 Forschungsmethodik . . . . .	5
1.5 Beitrag der Arbeit . . . . .	7
1.6 Gliederung der weiteren Arbeit . . . . .	8
<b>2 Programmierumgebungen in der Sekundarstufe I</b>	<b>11</b>
2.1 Algorithmik im Informatikunterricht . . . . .	11
2.1.1 Historische Entwicklung . . . . .	11
2.1.2 Aktuelle Sicht . . . . .	13
2.1.2.1 Internationale Standards . . . . .	14
2.1.2.2 Standards im deutschsprachigen Raum . . . . .	15
2.1.2.3 Gymnasiale Lehrpläne in Bayern . . . . .	16
2.2 Vermittlung von Konzepten der Algorithmik . . . . .	19
2.2.1 Übersicht . . . . .	19
2.2.2 Vermittlungskonzepte ohne Programmierumgebungen . . . . .	20
2.2.2.1 Alltagsprobleme . . . . .	20
2.2.2.2 Fachliche Problemstellungen . . . . .	20
2.2.3 Lern- und Programmierumgebungen . . . . .	21
2.2.3.1 Entwicklungsumgebungen für Programmiersprachen . . . . .	21
2.2.3.2 Ikonographische Programmierung . . . . .	23
2.3 Auswahl einer Lern- und Programmierumgebung . . . . .	26
2.3.1 Auswahlkriterien . . . . .	26
2.3.2 Analyse von Lern- und Programmierumgebungen . . . . .	28

2.3.3	Zusammenfassung und Fazit . . . . .	31
<b>3</b>	<b>Problemlösen</b>	<b>33</b>
3.1	Übersicht . . . . .	33
3.2	Problemlösestrategien . . . . .	34
3.2.1	Übersicht . . . . .	34
3.2.2	Problemlöseverfahren bei Programmieraufgaben . . . . .	37
3.3	Auswahl von Problemlösestrategien . . . . .	39
3.3.1	Untersuchungsrelevante Problemlösestrategien . . . . .	39
3.3.2	Einflussfaktoren für die Wahl bestimmter Strategien . . . . .	47
3.3.3	Aspekte aus psychologischer Sicht . . . . .	48
3.4	Zusammenfassung und Fazit . . . . .	48
<b>4</b>	<b>Systemrückmeldungen für Lernende</b>	<b>51</b>
4.1	Übersicht . . . . .	51
4.2	Variablen für Rückmeldungen . . . . .	54
4.2.1	Zeitpunkt . . . . .	55
4.2.2	Präsentation . . . . .	55
4.2.3	Gestaltung . . . . .	55
4.2.3.1	Ergebnisunabhängige Rückmeldung . . . . .	58
4.2.3.2	Ergebnisorientierte Rückmeldungen . . . . .	58
4.3	Zusammenfassung, Schlussfolgerung . . . . .	59
<b>5</b>	<b>Prozessbeobachtung</b>	<b>63</b>
5.1	Methoden der Prozessbeobachtung . . . . .	63
5.1.1	Protokollierung durch einen menschlichen Beobachter . . . . .	63
5.1.2	Rechnergestützte Beobachtung . . . . .	65
5.2	Auswertung von Prozessbeobachtungsdaten . . . . .	65
5.3	Zusammenfassung, Schlussfolgerungen . . . . .	67
<b>6</b>	<b>Forschungsfragen</b>	<b>69</b>
6.1	Automatisierte Identifizierung von Problemlösestrategien Lernender . . . . .	69
6.2	Erstellung von individualisierten Systemrückmeldungen für Lernende . . . . .	69
<b>7</b>	<b>Forschungsmethodik</b>	<b>71</b>
7.1	Voruntersuchungen . . . . .	71
7.2	Untersuchungsszenario der ersten Feldstudie . . . . .	72
7.2.1	Beschreibung der Gruppen . . . . .	72
7.2.2	Unterrichtsvorlauf . . . . .	72
7.2.3	Eingesetzte Aufgaben . . . . .	73
7.2.4	Untersuchungswerkzeuge . . . . .	73
7.3	Reliabilitäts- und Validitätsstudie . . . . .	74
7.3.1	Fragebögen nach Ajzen und Fishbein . . . . .	74

7.3.1.1	Theorie des <i>überlegten Handelns</i> und des <i>geplanten Verhaltens</i> . . . . .	75
7.3.1.2	Erstellung von Fragebögen . . . . .	76
7.3.1.3	Zusammenfassung, Wertung . . . . .	77
7.3.2	Methode des <i>laut Denkens</i> und qualitative Inhaltsanalyse . . .	78
7.3.2.1	Die Testpersonen . . . . .	78
7.3.2.2	Untersuchungsszenario . . . . .	78
7.4	Studie zur Datenerhebung für die Optimierung des Identifizierungswerkzeugs . . . . .	79
7.5	Untersuchungsszenario der zweiten Feldstudie . . . . .	79
7.5.1	Die Testpersonen . . . . .	79
7.5.2	Untersuchungsszenario . . . . .	80
7.5.3	Untersuchungswerkzeuge zur Datenanalyse . . . . .	80
7.6	Zusammenfassender Überblick . . . . .	81
<b>8</b>	<b>Ergebnisse</b> . . . . .	<b>85</b>
8.1	Ergebnisse der Voruntersuchungen . . . . .	85
8.2	Konzeption und Entwicklung der Untersuchungssoftware . . . . .	88
8.2.1	Softwareanforderungen an die Untersuchungssoftware . . . . .	88
8.2.2	Konzeption und Entwicklung der Beobachtungssoftware ( <i>TrackingKara</i> ) . . . . .	91
8.2.3	Konzeption und Entwicklung der Analyse- und Diagnosesoftware ( <i>EvalKara</i> ) . . . . .	94
8.2.3.1	Analyse des Lösungsverlaufs . . . . .	97
8.2.3.2	Analyse der Systemfehlermeldungen . . . . .	98
8.2.3.3	Analyse des Zeitaufwands für verschiedene Lösungsphasen . . . . .	99
8.2.3.4	Automatisierte Bewertung der Qualität der Lösungsversuche . . . . .	100
8.2.4	Softwarearchitektur . . . . .	103
8.3	Ergebnisse der ersten Feldstudie . . . . .	104
8.3.1	Erste empirische Ergebnisse . . . . .	104
8.3.2	Zwischenfazit . . . . .	108
8.3.3	Analyse der Chronologien der LSI . . . . .	109
8.3.4	Zusammenfassung der Analyseresultate . . . . .	117
8.3.5	Statistische Auswertungen zu Musterübergängen . . . . .	119
8.3.6	Zusammenfassung, Folgerungen . . . . .	124
8.4	Konzeption und Entwicklung der Strategiemustererkennungssoftware ( <i>IdentiKara</i> ) . . . . .	124
8.4.1	Zusammenhang zur Spracherkennung . . . . .	125
8.4.2	Erkennung von Mustern mittels Hidden-Markov-Modellen ( <i>HMM</i> ) . . . . .	125
8.4.3	Entwicklung geeigneter HMMs . . . . .	127
8.4.4	Konzeption der Softwarekomponente . . . . .	131
8.4.4.1	Anforderungen an die Erkennungssoftware . . . . .	132

8.4.4.2	Algorithmische Umsetzung des Erkennungsprozesses	133
8.4.4.3	Einbettung in die Lernumgebung . . . . .	136
8.5	Zusammenfassung . . . . .	138
8.6	Empirische Ergebnisse der zweiten Feldstudie . . . . .	140
8.6.1	Mittelwertsbetrachtungen . . . . .	144
8.6.2	Korrelationsuntersuchungen . . . . .	145
8.6.3	Regressionsanalyse und Varianzaufklärung . . . . .	146
8.6.4	Zusammenfassung . . . . .	148
8.7	Validierung der Zuordnung von Strategiemustern zu Problemlösestrategien . . . . .	148
8.7.1	Übersicht . . . . .	148
8.7.2	Methode des „lauten Denkens“ . . . . .	150
8.7.2.1	Datenerhebung . . . . .	151
8.7.2.2	Analyse der Daten . . . . .	153
8.7.2.3	Codierung der Daten . . . . .	154
8.7.2.4	Interpretation der Daten . . . . .	154
8.7.3	Diskussion der Resultate . . . . .	159
8.8	Gestaltung von individualisierten Systemrückmeldungen . . . . .	161
8.8.1	Teilaspekte der Rückmeldungen . . . . .	161
8.8.1.1	Informationen zur Lösungsqualität . . . . .	162
8.8.1.2	Informationen zur Fehlerkorrektur . . . . .	162
8.8.1.3	Motivierende Hinweise . . . . .	163
8.8.2	Gestaltung der Rückmeldungen und technische Umsetzung . . . . .	163
8.9	Übertragung der Konzepte auf andere Lern- und Programmierumgebungen . . . . .	169
8.9.1	Übersicht . . . . .	170
8.9.2	Übertragbarkeit auf andere Programmierumgebungen . . . . .	170
8.9.2.1	Übertragbarkeit auf visuelle Programmierumgebungen mit Codebausteinen . . . . .	171
8.9.2.2	Übertragbarkeit auf textuelle Programmierumgebungen didaktisch reduzierter Programmiersprachen . . . . .	176
8.9.2.3	Übertragbarkeit auf Programmierumgebungen gängiger Programmiersprachen . . . . .	178
8.9.3	Übertragbarkeit auf die integrierte Entwicklungsumgebung <i>BlueJ</i> . . . . .	179
8.9.4	Zusammenfassung . . . . .	180
<b>9</b>	<b>Zusammenfassung, Fazit und Ausblick</b>	<b>183</b>
9.1	Zusammenfassung . . . . .	183
9.1.1	Ausgangssituation . . . . .	183
9.1.2	Problemlöseprozess . . . . .	183
9.1.3	Prozessbeobachtung . . . . .	184
9.1.4	Automatisierte Identifizierung der Problemlösestrategien . . . . .	184
9.1.5	Individualisierte Systemrückmeldungen . . . . .	185
9.1.6	Übertragbarkeit der entwickelten Konzepte . . . . .	185

9.2	Fazit . . . . .	186
9.2.1	Beantwortung der Forschungsfragen . . . . .	186
9.2.2	Einsatzmöglichkeiten . . . . .	187
9.3	Offene Fragen und Ausblick . . . . .	188
<b>Literaturverzeichnis</b>		<b>208</b>
<b>Abbildungsverzeichnis</b>		<b>219</b>
<b>Tabellenverzeichnis</b>		<b>223</b>
<b>Verzeichnis der Programmlistings</b>		<b>225</b>
<b>Anhang</b>		<b>227</b>
A	Auswertung der Protokolle der Voruntersuchung . . . . .	228
B	Auswertung der ersten Feldstudie . . . . .	232
B.1	Empirische Daten . . . . .	232
B.2	LSI-Chronologien . . . . .	243
C	Auswertung der zweiten Feldstudie . . . . .	260
D	Fragebogen . . . . .	287
E	Protokolle „lautes Denken“ . . . . .	294
E.1	„Mädchen und Technik“-Praktikum 2009 – Stenogramme . . . . .	294
E.2	„Mädchen und Technik“-Praktikum 2010 – Stenogramme . . . . .	307
E.3	„Mädchen und Technik“-Praktikum 2009 – Transkripte . . . . .	316
E.4	„Mädchen und Technik“-Praktikum 2010 . . . . .	326
F	Auswertung der Validierungsstudie . . . . .	335
F.1	„Mädchen und Technik“-Praktikum 2009 . . . . .	335
F.2	„Mädchen und Technik“-Praktikum 2010 . . . . .	345
G	Beispiel für Testfälle einer komplexeren Kara-Aufgabe . . . . .	352
H	Systemrückmeldungen der Kara-Umgebung . . . . .	354
I	Textbausteine für individualisierte Systemrückmeldungen . . . . .	355
I.1	Textbausteine für <i>top down</i> -Strategie . . . . .	355
I.1.1	Informativer Anteil . . . . .	355
I.1.2	Motivationaler Anteil . . . . .	356
I.2	Textbausteine für <i>bottom up</i> -Strategie . . . . .	357
I.2.1	Informativer Anteil . . . . .	357
I.2.2	Motivationaler Anteil . . . . .	358
I.3	Textbausteine für <i>hill climbing</i> -Strategie . . . . .	359
I.3.1	Informativer Anteil . . . . .	359
I.3.2	Motivationaler Anteil . . . . .	360
I.4	Textbausteine für <i>trial and error</i> -Strategie . . . . .	361
I.4.1	Informativer Anteil . . . . .	361
I.4.2	Motivationaler Anteil . . . . .	362